Position Based Impedance Control of an Industrial Hydraulic Manipulator: Theory and Experiments

by

Bradley E.B. Heinrichs

98

A Thesis
Submitted to the Faculty of Graduate Studies
in Partial Fufillment of the Requirements
for the Degree of

Master of Science

Department of Mechanical and Industrial Engineering
University of Manitoba
Winnipeg, Manitoba

ISBN 0-612-13182-3

Name _____

Dissertation Abstracts International and *Masters Abstracts International* are arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation or thesis. Enter the corresponding four-digit code in the spaces provided.

Mechanical Engineering (Robotics)
SUBJECT TERM

0548
SUBJECT CODE

UMI

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS
Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION
General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523
Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS
Language
- General 0679
- Ancient 0289
- Linguistics 0290
- Modern 0291

Literature
- General 0401
- Classical 0294
- Comparative 0295
- Medieval 0297
- Modern 0298
- African 0316
- American 0591
- Asian 0305
- Canadian (English) 0352
- Canadian (French) 0355
- English 0593
- Germanic 0311
- Latin American 0312
- Middle Eastern 0315
- Romance 0313
- Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY
Philosophy 0422
Religion
- General 0318
- Biblical Studies 0321
- Clergy 0319
- History of 0320
- Philosophy of 0322

Theology 0469

SOCIAL SCIENCES
American Studies 0323
Anthropology
- Archaeology 0324
- Cultural 0326
- Physical 0327

Business Administration
- General 0310
- Accounting 0272
- Banking 0770
- Management 0454
- Marketing 0338

Canadian Studies 0385
Economics
- General 0501
- Agricultural 0503
- Commerce-Business 0505
- Finance 0508
- History 0509
- Labor 0510
- Theory 0511

Folklore 0358
Geography 0366
Gerontology 0351
History
- General 0578
- Ancient 0579
- Medieval 0581
- Modern 0582
- Black 0328
- African 0331
- Asia, Australia and Oceania 0332
- Canadian 0334
- European 0335
- Latin American 0336
- Middle Eastern 0333
- United States 0337

History of Science 0585
Law 0398
Political Science
- General 0615
- International Law and Relations 0616
- Public Administration 0617

Recreation 0814
Social Work 0452
Sociology
- General 0626
- Criminology and Penology 0627
- Demography 0938
- Ethnic and Racial Studies 0631
- Individual and Family Studies 0628
- Industrial and Labor Relations 0629
- Public and Social Welfare 0630
- Social Structure and Development 0700
- Theory and Methods 0344

Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES
Agriculture
- General 0473
- Agronomy 0285
- Animal Culture and Nutrition 0475
- Animal Pathology 0476
- Food Science and Technology 0359
- Forestry and Wildlife 0478
- Plant Culture 0479
- Plant Pathology 0480
- Plant Physiology 0817
- Range Management 0777
- Wood Technology 0746

Biology
- General 0306
- Anatomy 0287
- Biostatistics 0308
- Botany 0309
- Cell 0379
- Ecology 0329
- Entomology 0353
- Genetics 0369
- Limnology 0793
- Microbiology 0410
- Molecular 0307
- Neuroscience 0317
- Oceanography 0416
- Physiology 0433
- Radiation 0821
- Veterinary Science 0778
- Zoology 0472

Biophysics
- General 0786
- Medical 0760

EARTH SCIENCES
Biogeochemistry 0425
Geochemistry 0996
Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES
Environmental Sciences 0768
Health Sciences
- General 0566
- Audiology 0300
- Chemotherapy 0992
- Dentistry 0567
- Education 0350
- Hospital Management 0769
- Human Development 0758
- Immunology 0982
- Medicine and Surgery 0564
- Mental Health 0347
- Nursing 0569
- Nutrition 0570
- Obstetrics and Gynecology 0380
- Occupational Health and Therapy 0354
- Ophthalmology 0381
- Pathology 0571
- Pharmacology 0419
- Pharmacy 0572
- Physical Therapy 0382
- Public Health 0573
- Radiology 0574
- Recreation 0575
- Speech Pathology 0460
- Toxicology 0383

Home Economics 0386

PHYSICAL SCIENCES
Pure Sciences
Chemistry
- General 0485
- Agricultural 0749
- Analytical 0486
- Biochemistry 0487
- Inorganic 0488
- Nuclear 0738
- Organic 0490
- Pharmaceutical 0491
- Physical 0494
- Polymer 0495
- Radiation 0754

Mathematics 0405
Physics
- General 0605
- Acoustics 0986
- Astronomy and Astrophysics 0606
- Atmospheric Science 0608
- Atomic 0748
- Electronics and Electricity 0607
- Elementary Particles and High Energy 0798
- Fluid and Plasma 0759
- Molecular 0609
- Nuclear 0610
- Optics 0752
- Radiation 0756
- Solid State 0611

Statistics 0463

Applied Sciences
Applied Mechanics 0346
Computer Science 0984
Engineering
- General 0537
- Aerospace 0538
- Agricultural 0539
- Automotive 0540
- Biomedical 0541
- Chemical 0542
- Civil 0543
- Electronics and Electrical 0544
- Heat and Thermodynamics 0348
- Hydraulic 0545
- Industrial 0546
- Marine 0547
- Materials Science 0794
- Mechanical 0548
- Metallurgy 0743
- Mining 0551
- Nuclear 0552
- Packaging 0549
- Petroleum 0765
- Sanitary and Municipal 0554
- System Science 0790

Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY
General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

Nom _____

Dissertation Abstracts International est organisé en catégories de sujets. Veuillez s.v.p. choisir le sujet qui décrit le mieux votre thèse et inscrivez le code numérique approprié dans l'espace réservé ci-dessous.

_____ ☐☐☐ U·M·I

SUJET — CODE DE SUJET

Catégories par sujets

HUMANITÉS ET SCIENCES SOCIALES

COMMUNICATIONS ET LES ARTS
Architecture 0729
Beaux-arts 0357
Bibliothéconomie 0399
Cinéma 0900
Communication verbale 0459
Communications 0708
Danse 0378
Histoire de l'art 0377
Journalisme 0391
Musique 0413
Sciences de l'information 0723
Théâtre 0465

ÉDUCATION
Généralités 515
Administration 0514
Art 0273
Collèges communautaires 0275
Commerce 0688
Économie domestique 0278
Éducation permanente 0516
Éducation préscolaire 0518
Éducation sanitaire 0680
Enseignement agricole 0517
Enseignement bilingue et multiculturel 0282
Enseignement industriel 0521
Enseignement primaire. 0524
Enseignement professionnel 0747
Enseignement religieux 0527
Enseignement secondaire 0533
Enseignement spécial 0529
Enseignement supérieur 0745
Évaluation 0288
Finances 0277
Formation des enseignants 0530
Histoire de l'éducation 0520
Langues et littérature 0279
Lecture 0535
Mathématiques 0280
Musique 0522
Orientation et consultation 0519
Philosophie de l'éducation 0998
Physique 0523
Programmes d'études et enseignement 0727
Psychologie 0525
Sciences 0714
Sciences sociales 0534
Sociologie de l'éducation 0340
Technologie 0710

LANGUE, LITTÉRATURE ET LINGUISTIQUE
Langues
- Généralités 0679
- Anciennes 0289
- Linguistique 0290
- Modernes 0291

Littérature
- Généralités 0401
- Anciennes 0294
- Comparée 0295
- Médiévale 0297
- Moderne 0298
- Africaine 0316
- Américaine 0591
- Anglaise 0593
- Asiatique 0305
- Canadienne (Anglaise) 0352
- Canadienne (Française) 0355
- Germanique 0311
- Latino-américaine 0312
- Moyen-orientale 0315
- Romane 0313
- Slave et est-européenne 0314

PHILOSOPHIE, RELIGION ET THÉOLOGIE
Philosophie 0422
Religion
- Généralités 0318
- Clergé 0319
- Études bibliques 0321
- Histoire des religions 0320
- Philosophie de la religion 0322

Théologie 0469

SCIENCES SOCIALES
Anthropologie
- Archéologie 0324
- Culturelle 0326
- Physique 0327

Droit 0398
Économie
- Généralités 0501
- Commerce-Affaires 0505
- Économie agricole 0503
- Économie du travail 0510
- Finances 0508
- Histoire 0509
- Théorie 0511

Études américaines 0323
Études canadiennes 0385
Études féministes 0453
Folklore 0358
Géographie 0366
Gérontologie 0351
Gestion des affaires
- Généralités 0310
- Administration 0454
- Banques 0770
- Comptabilité 0272
- Marketing 0338

Histoire
- Histoire générale 0578
- Ancienne 0579
- Médiévale 0581
- Moderne 0582
- Histoire des noirs 0328
- Africaine 0331
- Canadienne 0334
- États-Unis 0337
- Européenne 0335
- Moyen-orientale 0333
- Latino-américaine 0336
- Asie, Australie et Océanie 0332

Histoire des sciences 0585
Loisirs 0814
Planification urbaine et régionale 0999
Science politique
- Généralités 0615
- Administration publique 0617
- Droit et relations internationales 0616

Sociologie
- Généralités 0626
- Aide et bien-être social 0630
- Criminologie et établissements pénitentiaires 0627
- Démographie 0938
- Études de l' individu et de la famille 0628
- Études des relations interethniques et des relations raciales 0631
- Structure et développement social 0700
- Théorie et méthodes. 0344
- Travail et relations industrielles 0629

Transports 0709
Travail social 0452

SCIENCES ET INGÉNIERIE

SCIENCES BIOLOGIQUES
Agriculture
- Généralités 0473
- Agronomie. 0285
- Alimentation et technologie alimentaire 0359
- Culture 0479
- Élevage et alimentation 0475
- Exploitation des pâturages 0777
- Pathologie animale 0476
- Pathologie végétale 0480
- Physiologie végétale 0817
- Sylviculture et faune 0478
- Technologie du bois 0746

Biologie
- Généralités 0306
- Anatomie 0287
- Biologie (Statistiques) 0308
- Biologie moléculaire 0307
- Botanique 0309
- Cellule 0379
- Écologie 0329
- Entomologie 0353
- Génétique 0369
- Limnologie 0793
- Microbiologie 0410
- Neurologie 0317
- Océanographie 0416
- Physiologie 0433
- Radiation 0821
- Science vétérinaire 0778
- Zoologie 0472

Biophysique
- Généralités 0786
- Medicale 0760

SCIENCES DE LA TERRE
Biogéochimie 0425
Géochimie 0996
Géodésie 0370
Géographie physique 0368
Géologie 0372
Géophysique 0373
Hydrologie 0388
Minéralogie 0411
Océanographie physique 0415
Paléobotanique 0345
Paléoécologie 0426
Paléontologie 0418
Paléozoologie 0985
Palynologie 0427

SCIENCES DE LA SANTÉ ET DE L'ENVIRONNEMENT
Économie domestique 0386
Sciences de l'environnement 0768
Sciences de la santé
- Généralités 0566
- Administration des hôpitaux 0769
- Alimentation et nutrition 0570
- Audiologie 0300
- Chimiothérapie 0992
- Dentisterie 0567
- Développement humain 0758
- Enseignement 0350
- Immunologie 0982
- Loisirs 0575
- Médecine du travail et thérapie 0354
- Médecine et chirurgie 0564
- Obstétrique et gynécologie 0380
- Ophtalmologie 0381
- Orthophonie 0460
- Pathologie 0571
- Pharmacie 0572
- Pharmacologie 0419
- Physiothérapie 0382
- Radiologie 0574
- Santé mentale 0347
- Santé publique 0573
- Soins infirmiers 0569
- Toxicologie 0383

SCIENCES PHYSIQUES
Sciences Pures
Chimie
- Généralités 0485
- Biochimie 487
- Chimie agricole 0749
- Chimie analytique 0486
- Chimie minérale 0488
- Chimie nucléaire 0738
- Chimie organique 0490
- Chimie pharmaceutique 0491
- Physique 0494
- PolymÈres 0495
- Radiation 0754

Mathématiques 0405
Physique
- Généralités 0605
- Acoustique 0986
- Astronomie et astrophysique 0606
- Électronique et électricité 0607
- Fluides et plasma 0759
- Météorologie 0608
- Optique 0752
- Particules (Physique nucléaire) 0798
- Physique atomique 0748
- Physique de l'état solide 0611
- Physique moléculaire 0609
- Physique nucléaire 0610
- Radiation 0756

Statistiques 0463

Sciences Appliqués Et Technologie
Informatique 0984
Ingénierie
- Généralités 0537
- Agricole 0539
- Automobile 0540
- Biomédicale 0541
- Chaleur et thermodynamique 0348
- Conditionnement (Emballage) 0549
- Génie aérospatial 0538
- Génie chimique 0542
- Génie civil 0543
- Génie électronique et électrique 0544
- Génie industriel 0546
- Génie mécanique 0548
- Génie nucléaire 0552
- Ingénierie des systèmes 0790
- Mécanique navale 0547
- Métallurgie 0743
- Science des matériaux 0794
- Technique du pétrole 0765
- Technique minière 0551
- Techniques sanitaires et municipales 0554
- Technologie hydraulique 0545

Mécanique appliquée 0346
Géotechnologie 0428
Matières plastiques (Technologie) 0795
Recherche opérationnelle 0796
Textiles et tissus (Technologie) 0794

PSYCHOLOGIE
Généralités 0621
Personnalité 0625
Psychobiologie 0349
Psychologie clinique 0622
Psychologie du comportement 0384
Psychologie du développement 0620
Psychologie expérimentale 0623
Psychologie industrielle 0624
Psychologie physiologique 0989
Psychologie sociale 0451
Psychométrie 0632

POSITION BASED IMPEDANCE CONTROL OF AN INDUSTRIAL HYDRAULIC MANIPULATOR: THEORY AND EXPERIMENTS

BY

BRADLEY E.B. HEINRICHS

A Thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

Abstract

This thesis documents the development of position based impedance control for implementation on an industrial hydraulic manipulator. The implementation is made possible by a novel Nonlinear PI (NPI) controller. The NPI controller is demonstrated to improve position tracking accuracy on an existing industrial hydraulic manipulator by a factor of five relative to an optimized linear PI controller, without sacrificing regulation accuracy or robustness. The position based impedance controlled manipulator, incorporating the NPI controller, is shown to be capable of motion in free space, static and dynamic force control, and controlled environmental collisions in both flexible and stiff environments.

Theoretical analysis shows that the position based formulation of impedance control has different force control behaviour from the torque based formulation. Experimental work confirms a new relationship, derived here, between explicit force control and position based impedance control. Further theoretical work shows that the existence of any steady-state positioning error will lead to recurring oscillatory behaviour if the static component of the target impedance is less than twice the stiffness of the combined environment and force sensor. Experiments demonstrate this limitation, and show that application-specific compensation methods can be used to decrease the apparent manipulator stiffness without compromising stability.

Acknowledgements

I would like to thank Dr. Sepehri and Dr. Thornton-Trump for giving me the opportunity to work on this project, and for their help in reviewing and editing this work. Al Lohse spend much time instumenting and tuning the Unimate robot before I ever worked on it, and his work made this one possible. I am very grateful to NSERC for their financial support of my studies. Finally, I would like to thank my wife Stephanie for her encouragement to finish this quickly.

Table of Contents

List of Figures

1. Introduction

1.1 Objectives

The field of robotics has seen rapid growth and development in the last two decades. Robots have been introduced into industry with great success. Applications such as welding and spray painting have been especially well suited to robots because they do not require environmental interaction. For applications such as these, it is sufficient for the manipulator to track a predetermined trajectory in space. However, there is a large class of potential robotic applications requiring environmental interactions. Reaming, boring, drilling, routing, grinding, bending, scraping and chipping are all examples of such tasks. The position based controllers that work so well for painting and welding take no account of forces due to environmental interaction. They are therefore ineffective and even dangerous in these applications. What is needed is a method to control the forces of interaction between the manipulator and the environment. The goal of this work is to develop a formulation of position based impedance control (PBIC) for implementation on an existing industrial hydraulic manipulator. The performance of the controlled manipulator in a set of tasks will be documented under a variety of conditions.

1.2 Background of the Problem

About 15 years ago, advances began to be made in the area of force control. The following chapter presents an overview of the work that has been done in this area. The fruits of this work are then examined in terms of their applicability to industrial hydraulic manipulators. First, however, some terminology and the nature of the general force control problem will be explained.

In general, a manipulator consists of a series of rigid bodies referred to as links. Adjacent links in the chain are connected by joints, each joint usually allowing one degree-of-freedom, either rotational or translational. The point at the end of the manipulator is referred to as the end-effector, being a gripper or some other implement. The Cartesian position and orientation of the end-effector is represented as the vector $\mathbf{x}$, and is related to

the set of generalized joint coordinates $\mathbf{q}$ through kinematic relations. The Jacobian matrix $\mathbf{J}$ providing the differential relationship between these quantities, i.e. $\delta\mathbf{x}=\mathbf{J}\delta\mathbf{q}$. The same matrix relates the Cartesian forces at the end-effector, $\mathbf{f}$, to equivalent generalized joint torques, τ, through its transpose: $\tau = \mathbf{J}^T\mathbf{f}$.

The fundamental problem facing force control is this: force and position are conjugate variables, and therefore both cannot be specified simultaneously. This constraint is easily recognized from simple Newtonian mechanics; from Newton's second law, to simultaneously control force and position (and, through its derivatives, acceleration) would be to control the physical mass of the system, which is impossible. Therefore, simultaneous control of collinear positions and forces is also impossible. Position based impedance control is one method of trading off the control of position and force in a way that is consistent with this reality.

There are a number of challenging practical considerations which are particular to this work specifically. These difficulties stem from the desire to apply the controller developed herein to an existing industrial hydraulic manipulator possessing many nonlinearities and nonidealities. Two such practical considerations, Coulomb friction and actuator deadband present particularly troublesome impediments to successful implementation of any control strategy.

1.3 Outline of the Thesis

This work is developed as follows:

1) A review of the literature relating to the objectives of this work, as outlined above, is presented. This review will suggest that a position based formulation of impedance control is particularly well-suited to the problem of controlling industrial hydraulic manipulators (Chapter 2).

2) The theoretical underpinnings of impedance control are discussed, and the controller structure is given (Chapter 3).

3) A nonlinear PI position controller is developed for use with the impedance controller. Each step in the development is experimentally documented, showing the progression from a simple linear PI controller to a robust and accurate nonlinear controller (Chapter 4).

4) The impedance controller is tested by a wide range of tasks and under very different environmental conditions (Chapter 5).

5) Some fundamental limitations of the position based impedance control technique are derived and demonstrated experimentally. Techniques to circumvent these limitations are suggested and tested. The relationship between the position based formulation of impedance control and the more common torque based formulation is derived and discussed, showing previously unremarked differences between the techniques (Chapters 6 & 7).

6) Conclusions about the merits of the position based impedance control technique are presented, along with recommendations on areas for future improvement and exploration (Chapters 8 & 9).

2. Background

As already stated, the first developments in robotic control involved the control of manipulator motions in the complete absence of environmental interaction, i.e. position control. Some of the developments in this area are reviewed because they form the basis for later force-control techniques, of which impedance control is one technique.

2.1 Position Control

2.1.1 Linear Position Control

The simplest way to implement a position controller is to transform the desired end-effector trajectory into joint space using the inverse kinematic solution for the manipulator (see Appendix A), then to use a PID type controller on each joint. Discussions of these controllers can be found in any undergraduate controls textbook (e.g. D'Souza, 1988). More sophisticated controllers form the basis for many force control strategies. These more advanced controllers will be reviewed below.

2.1.2 Computed Torque Technique

R.C. Paul (1972) suggested the feasibility of using a dynamic model of a manipulator to compute the necessary torques to achieve a desired set of joint accelerations. This torque could then be used in a feedforward loop to improve the manipulator response. In particular, the control system would be able to compensate for nonlinear torques due to gravity, viscous friction, centripetal and Coriolis forces. Supplemental accelerations were added based on the remaining joint position and velocity errors. Then, given accurate modeling, these supplemental accelerations would produce a second-order rejection of joint angle positioning errors.

Although originally conceived of as an improved method of position control, it formed the basis for later attempts to synthesize a force-control strategy. It should be noted at this point that the feedforward signal developed in the computed torque technique is evidently a torque, and is therefore best suited to electrically-actuated robots,

since electric motors can easily produce torques in direct proportion to their control signals.

2.1.3 Resolved Acceleration Control

In 1980, Luh, Walker, and Paul presented the concept of resolved acceleration control. Again conceived of as an augmented position trajectory tracker, this approach differed from the computed torque technique in that the desired vector of joint accelerations was modified by considerations of the Cartesian errors in end-point position, velocity and orientation errors. Orientation errors were represented by a sum of the cross product of the end-point orientation vectors with the desired quantities. The control torques at each joint were then calculated as in the computed torque technique. In this way, for a good dynamic model of the manipulator, the position and orientation of the end-effector should exhibit a second-order rejection of disturbances, compared with a rejection at the joint level obtained in the computed torque technique. Although not originally noticed, the move from a joint-space controlled response to a Cartesian-space controlled response made the resolved acceleration technique applicable to force control, since the forces to be regulated occur in the Cartesian workspace. Again, it is noted that the output of this control algorithm is a set of control torques, making it particularly suitable to electric robots.

2.2 Explicit Force Control

The antithesis of pure position control is pure force control. While it is the objective of this work to control more than just force, the work done in force control relating to hydraulic systems is well worth reviewing. The focus of this section will be confined to work of direct relevance to hydraulic systems.

Conrad and Jensen (1987) addressed the problem of force regulation in a single hydraulic piston. Combinations of velocity feedforward, control signal feedback and a Luenberger observer and state estimate feedback were implemented in simulation (using a third-order linear model) and experiment. Comparison of the results showed that using all

three methods in conjunction provided the most significantly improved response relative to simple PI feedback control. This improvement was observed in both steady-state error and disturbance rejection. It is important to note, however, that stiction was eliminated in the experimental apparatus with special fluorocarbon seals.

Pannala et al. (1989) used multivariable frequency-response techniques to reduce undesirable interactions between actuators in industrial hydraulic machinery. The experimental apparatus consisted of a cantilever beam operated upon at two points by two different hydraulic actuators. Controller gains were empirically determined using an "Interactive Graphics Optimization Procedure". In a task consisting of a step increase in the force output of one actuator, the controller was shown to be effective in reducing actuator interactions. The regulation abilities of the combined system were not discussed.

Chen et al. (1990) designed a variable structure controller with sliding mode control for a single hydraulic piston. Using position, velocity, acceleration, force and piston pressure feedback, the variable structure controller proved to be highly capable in both static and dynamic force control tasks. Reference force signal tracking was demonstrated at up to 2_{Hz}. The details of the physical system upon which the experiments were performed were not given, but the assumption of a compliant environment (relative to the piston and sensor) was stated.

2.3 Hybrid Force/Position Control

In 1981, Raibert and Craig proposed a way of dealing with the problem of controlling position while dealing with constraints in the workspace. They claimed that any manipulation task could be divided into two complementary spaces. Because it is physically impossible to specify both force and position simultaneously along any one degree-of-freedom, each quantity should be controlled only in its own subspace. For example, explicit force control (meaning not relying on an underlying position controller) could be applied along the Cartesian X-axis, and position control could be applied along the orthogonal Y and Z axes.

This control method is accomplished by defining a selection matrix, **S**, indicating the degrees of freedom along which the force is to be controlled. In the above example, **S** would have a 1 in the top left corner, indicating force control in X, the rest consisting of zeroes, indicating position control in Y and Z. The appropriate set of control torques is obtained by adding the torques required by the force controller weighted by **S** to those required by the position controller weighted by **I-S** (**I** being the identity matrix). The selection matrix must be determined by some higher-level supervisor, using either knowledge about the workspace or some kind of adaptive feedback scheme.

The output of the control algorithm proposed by Raibert and Craig is a set of control torques. As such, hybrid control has been used extensively with direct-drive electric manipulators (An and Hollerbach, 1987a-b; Dawson et al., 1992; Volpe and Khosla, 1993a-b).

Joint space formulations of hybrid control have since been shown to be possible by transforming the selection matrix into joint space (Zhang and Paul, 1985). This formulation, however, as that proposed but Raibert and Craig, exhibits kinematic instabilities, even for well-conditioned manipulator Jacobians (An and Hollerbach, 1987b). A corrected, stable formulation has since been proposed by Fisher and Mujtaba (1992) which extends the applicability of the technique to redundant manipulators.

Volpe and Khosla (1993a-b) have presented a number of thorough experimental investigations of the performance of various controllers on a set of standardized tasks. One such study (1993a) contrasted the ability of various explicit force controllers to follow a set of force steps and ramps. The equivalence of explicit force controllers and torque based implementations of impedance control was shown. The controller was implemented on a six degree-of-freedom direct-drive arm. The environment used for most tests consisted of a steel plate atop a cardboard box, although the more promising controllers were also tested on a rigid steel environment. No passive compliance was used in the force sensor. Tests of all subsets of PID control acting on the force error showed that integral action provided the most effective force trajectory tracking. Actually, it was

concluded that a second-order low pass filter (double integral control action) provided even better force tracking, though only slightly.

Another study by Volpe and Khosla (1993b) centered on the performance of various controllers with respect to large changes in environmental admittance (impact with a stiff environment). A review of existing impact control strategies showed that active damping was problematic, passive compliance resulted in a reduction of manipulator capabilities, and integral force control was ineffective in high-energy impacts due to integral wind-up. Using the same experimental setup as Volpe and Khosla (1993a), it was shown that contact instabilities could be almost eliminated by reducing proportional force feedback gains to between 0 and -1. Using the relation first shown by Hogan (1987), it is apparent that this choice of feedback gains is equivalent to greatly increasing the target inertia when using a torque based impedance controller (see Section 6.1). A further finding was that, while this gain selection improved contact, large steady-state errors ensued. It was therefore concluded that these control gains should only be used in the contact phase, lasting about 0.15 seconds. After this period, a linear switch of all controller gains was proposed, over another 0.15 second period. This technique was shown to result in rapid adjustment to large changes in environmental admittance and no steady-state error in force regulation.

Tsai and Orin (1986) documented the application of a modified hybrid controller to a hydraulically-actuated hexapodal walking vehicle. Instead of closing the control loop at the end-effector, the loop was closed at the joint level by transforming the selection matrix, **S,** to joint space using the manipulator Jacobian. The controlled system was demonstrated to be capable of orthogonal position and force control in a simulated walking task. This level of performance was possible because the actuators were high-speed and could be modeled by a linearized inverse dynamic module for feedforward purposes.

2.4 Stiffness Control

Salisbury (1980) proposed a simple alternative method for dealing with forces at the end-effector. A stiffness relationship was specified between the Cartesian position of the end-effector and the forces acting upon it. This relationship was translated into joint space, and was used to compute the actuator torques necessary to have the end-effector display the desired behaviour. The technique was demonstrated to be effective in limiting forces when implemented on an electric robot whose task was to insert a peg into a hole. This work is a precursor to a more general approach, called impedance control.

2.5 Impedance Control

Hogan (1985a-c) presented a three-part paper detailing the theory, implementation and applications of a novel control method which he termed "impedance control". This method was presented as a general and unified approach to all of manipulation. The fundamental idea of impedance control is that it is not the forces on the end-effector that should be controlled, but rather its apparent dynamic response. The apparent robot dynamics, or impedance, should then be modulated to obtain the desired forces and positions at the end-effector. The technique can therefore be seen as a generalization of stiffness control (see Salisbury, 1980). The fundamentals of impedance control as developed by Hogan in this seminal paper will be described in Section 3. Some of the relevant subsequent work in the area is described below.

2.5.1 Stability

Stability analyses confirmed the relationship between force-feedback gains and effective inertia found by Hogan (see An and Hollerbach, 1987b; Volpe and Khosla, 1993b; see also Section 6.1). An and Hollerbach (1987b) reasoned that any system involving force feedback from a wrist mounted force sensor must be run at a slow rate because the sensor is mounted at the end of a low-bandwidth flexible system. They therefore suggested the use of fast servo torque control loops with a slow outer loop involving force-feedback from the end-effector. They further demonstrated the

effectiveness of using dominant pole filtering on the control torques. The method allowed the use of much higher stiffnesses in the target impedance without inducing oscillations. This was demonstrated experimentally using a single link of the MIT Direct-Drive arm.

Investigation of kinematic stability issues showed that stiffness control and resolved acceleration control (assumed to be subsets of impedance control) did not exhibit kinematic instabilities, but a hybrid controller did (An and Hollerbach, 1987a). Furthermore, the resolved acceleration controller proved robust to large modeling errors. These results were found using simulation and experiment on a two-link direct-drive articulated arm. The instabilities observed in hybrid control were attributed to interaction between the manipulator inertia matrix inverse and the inverse manipulator Jacobian. As already mentioned, Fisher and Mujtaba (1992) have since proposed a corrected, kinematically stable formulation of hybrid control.

2.5.2 Hybrid Impedance Control

Goldenberg suggested a hybrid impedance controller for improved force trajectory tracking (Goldenberg, 1991). The approach taken was exactly that of the hybrid position/force control technique (Raibert and Craig, 1981), but impedance control was implemented in the subspace where position control would otherwise be used. A desired acceleration trajectory was found, and used to compute the control torques as in the resolved acceleration method. This method was implemented on a two degree-of-freedom direct-drive manipulator. It proved to be able to accurately track a 1Hz sine wave force trajectory, and was robust to large modeling errors. It also continued stable operation after sudden contact with an aluminum wall.

2.5.3 Adaptive Impedance Control

The fundamental task of the impedance controller, as laid out by Hogan (1985), is to make the manipulator dynamics conform to the target impedance. This impedance should then be selected and modulated as required by the task at hand. Two approaches have been taken in addressing the later issue. The first is to change the form or parameters in the target impedance. The second is to alter the reference trajectory.

Lu and Meng (1991) examined the performance of two adaptive implementations of impedance control. They found that the injection of adaptive control techniques could compensate for errors in the dynamic model of the manipulator, resulting in more robust control. These conclusions were arrived at by Lyapunov analysis and simulation of a two-link direct-drive manipulator.

Cohen and Flush (1991) examined the problem designing a higher-level controller where the controlled variables were the target dynamics of the manipulator. Studies were carried out on a simulation of a two-link direct-drive manipulator performing some sort of wiping task. The impedance model used was that of a linear simple impedance, a term defined by Hogan (1988) meaning an impedance of the form $\mathbf{f} = \mathbf{A}\dot{\mathbf{x}} + \mathbf{B}\mathbf{x}$. An associative search network was used to learn impedance parameters using a reward function involving the sum of absolute errors in velocity and force. The errors were taken with respect to reference trajectories. Another important element of this particular work was that no feedforward torques were used. The commanded torques were simply those required by the impedance relation in the static case, using a fixed inertial matrix. It was found that the impedance parameter modulation scheme was fairly successful, but was unable to deal with abrupt changes in environmental admittance. A reference-trajectory modification scheme was implemented in an attempt to deal with this problem, with the result that stability and trajectory tracking were both improved.

Another study of impedance parameter modulation (Johansson and Spong , 1994) used a quadratic optimization scheme to minimize deviations from velocity and force trajectories. Again, a linear simple impedance was used on a simulation of a two-link direct-drive robot. However, the authors only considered constant interaction forces and, not surprisingly, the results showed great stability.

Lasky and Hsia (1991) examined the usefulness of further modifying the reference trajectory for the purpose of force tracking. Their approach was to modify each coordinate of the reference trajectory in direct proportion to the error in the corresponding component of the end-effector force. The effectiveness of the proposed modification was tested on a simulation of a two-link direct-drive manipulator with a second-order target

impedance. The task was to deburr a surface, which was simulated as a pure stiffness. The results contrasted the performance with and without trajectory modification, showing much improved force tracking when the proposed modification was used. This modification proved robust to multiple order-of-magnitude errors in estimation of environmental stiffness. However, contact was assumed throughout the simulation, and therefore impact dynamics were not addressed.

2.5.4 Force Regulation using Impedance Control

Using impedance control as a method of force trajectory tracking has been the focus of many other papers. One suggested means of controlling both force and system dynamics was to add a reference force to the force required by the target impedance (Goldenberg, 1988). This is equivalent to adding another impedance in parallel with the target impedance. The superposition property (Hogan, 1985a) of impedance control makes such an approach permissible. The idea was not experimentally tested, however.

McCormick and Stewards (1993) compared the results of using a linearized impedance controller with a fully nonlinear one in terms of force trajectory tracking and contact stability. A second-order target impedance was chosen, and the force offset method of Goldenberg (1988) was used to specify the reference force trajectory. Using a direct-drive, five bar linkage, a slightly better response was observed for the nonlinear controller. However, simulation had shown kinematic instabilities in the nonlinear controller that were absent in its linearized counterpart. Contact studies showed that, for improper impedance parameter selection, instability was present even when the environment's steel surface was covered with a rubber pad. Increasing the target inertia, and other parameters proportionately, resulted in stable contact.

Dawson, Lewis and Dorsey (1992) have also applied impedance control to force tracking tasks. They compensated for errors in the dynamic model of the manipulator using a Lyapunov based auxiliary signal to augment a PD based position controller. Given controllable actuator torques and exact kinematic information, the global ultimate

boundedness of the tracking error was proven, and simulation showed good force regulation.

2.6 Position Based Impedance Control (PBIC)

It is apparent at this point that almost all the implementations and experiments reviewed thus far have involved direct-drive electrical manipulators. The fundamental assumption undergirding these works is that the manipulator to be controlled is equipped with actuators that exert controllable torques (Hogan, 1987). This is the basis for the use of all the methods which incorporate a computed torque controller, and is certainly crucial for any conclusions about stability. This assumption of controllable torques is good for electrically-actuated manipulators with little gearing backlash. It is not suitable, however, for hydraulically actuated robots.

An electric actuator exerts a torque which is directly proportional to the control current (D'Souza, 1988). In a hydraulic actuator, the control current acts to move the spool valve, which controls the flow of hydraulic fluid into the piston. This flow in turn causes a pressure build-up, which is proportional to the actuator torque. Even if the spool valve dynamics are ignored, the control current fundamentally controls the derivative of the actuator torque, not the torque itself. There is therefore automatically some lag introduced into the system. Even if this lag were small compared with the target dynamics, the problem of controlling the pressure difference in the piston is a difficult problem (Conrad and Jensen, 1987). Therefore, what is needed to meet the stated objectives of this work is an implementation of impedance control that does not require controlled actuator torques. Position based impedance control is such a method.

Position based (or model reference) impedance control (see Field and Stepanenko, 1993) is a position controller nested within a force feedback loop. Force feedback is used in a digital model of the target impedance to modify the position set-point. Thus, unlike torque based implementations of impedance control (TBIC), this implementation requires no dynamic model of the robot or the environment, simplifying and speeding control.

Position based impedance control has previously been implemented on a six degree-of-freedom electric manipulator (Field and Stepanenko, 1993). This manipulator did not fit the requirements of the standard impedance control implementation because of lack of good dynamic modeling and a non-backdrivable transmission. It was demonstrated that the control system could exhibit the target dynamic behaviour under tests of step changes in position or end-effector loading. However, stability problems were observed when sudden environmental contact was made. Attempts to deal with these instabilities involved increased damping in the target dynamics, passive sensor compliance, and damping parameter adaptation based on environmental knowledge. The latter approaches met with the most success, damping out oscillations within 2 seconds. No investigation was made on the effects of increasing the target inertia, as suggested by Volpe and Khosla (1993b).

2.7 Significance of the Current Work

The current work fits into and builds upon the body of literature just reviewed by:

1) developing the theory of position based impedance control as it relates to any robot, not just industrial hydraulic manipulators, by deriving new relations showing differences between the force control behaviours of torque and position based implementations of impedance control,

2) analytically demonstrating some fundamental limitations of position based impedance control,

3) providing experimental verification of both of the above theoretical results,

4) developing a robust and accurate position controller for an existing industrial hydraulic manipulator, and

5) for the first time, implementing and assessing impedance control on the same existing industrial hydraulic manipulator.

These contributions are developed in detail in the remaining chapters of this thesis.

3. Theory

3.1 The Fundamentals of Impedance Control

As shown in the review of the various approaches to force control prior to 1985, tasks were divided into degrees of freedom along which different control methods were applied. Hogan (1985a) stated that what was needed was not a set of techniques specialized to each individual task, but a unified and general approach to all of manipulation. He began the theoretical development of his approach with the following postulate:

"It is impossible to devise a controller which will cause a physical system to present an apparent behaviour to its environment which is distinguishable from that of a purely physical system."

This reasonable postulate was used as the basis for applying the laws of causality that govern physical systems.

At the juncture of any two physical systems, instantaneous power flow is the product of two conjugate variables. For example, electrical power is the product of voltage and current. Hydraulic power is the product of pressure and volumetric flow rate. Mechanical power is the product of force and velocity. In each case, the power is the product of an effort and a flow. Furthermore, no physical system can determine both of these variables (the recognition of this latter fact is the basis for using hybrid control). Along a given degree-of-freedom, any physical system accepts one of these variables as an input, and produces the other as an output. A system accepting an effort and producing a

flow is termed an admittance, while one that accepts flow and produces effort is an impedance.

Hogan impressed on the reader the fundamental distinction between impedance and admittance. He argued that, while the one is merely the inverse of the other in linear systems, the distinction is paramount if the system is nonlinear. For example, the phenomenon of stiction must be viewed as a force resulting from an imposed velocity; the relationship cannot be inverted. Conversely, the velocity of a point mass is the result of the integral of the time history of the forces imposed upon it. Physical systems must be properly described in one form or the other.

The task then is to determine which form suits which half of the robot/environment system. Because the manipulator is the object under control, its behaviour should be matched to be conjugate to that of the environment, i.e. an impedance should be matched to an admittance, and vice versa. The question, then, is whether the environment is best described as an impedance or as an admittance.

Hogan argued that an admittance is the correct representation of the environment. He compared the object to be manipulated to a possibly constrained inertial mass. If the mass is unconstrained, the mass accepts a force and produces a velocity, as described above. If constrained, the behaviour of the mass is even more firmly admissive; it can be pushed, but it may not move at all. Under this line of reasoning, the correct description of the environment is as an admittance. Therefore, the correct behaviour that should be imposed on the manipulator is that of an impedance.

3.1.1 Implementation

The implementation of impedance control suggested by Hogan is conceptually very similar to stiffness control. The difference lies in the manner in which the desired accelerations are derived. In impedance control, the accelerations are the superposition of those needed to follow a specified position trajectory and those required by the desired impedance relation and forces on the end-effector. Effectively, feedforward torques are used to cancel the actual manipulator dynamics and replace them with the desired relations

between force and velocity. As in the resolved acceleration method, accelerations are calculated at the end-effector, and in Cartesian space.

3.1.2 Impedance Selection

A difference between force control and impedance control is that the impedance controller does not see dynamic interactions as a disturbance to be rejected, but rather as integral to the nature of the control task. The impedance controller attempts to modulate these dynamics to control the behaviour of the complete system. Hogan developed some rudimentary guidelines as to how this modulation should be carried out.

Hogan analyzed two types of tasks where the manipulator interacts with the environment. The first task was to transfer power to the environment. He considered only passive environments and a single degree-of-freedom, but claimed that the basic structure of the problem was preserved. He found that, to maximize power transfer, the manipulator's impedance should be matched to that of the environment. More correctly, the product of the target impedance and the environmental admittance should be unity. This result is consistent with the familiar rule for maximum power transfer in electrical circuits.

The other task analyzed was to minimize deviations from a set trajectory while simultaneously minimizing interaction forces. Optimizing a quadratic cost function, Hogan found that the impedance chosen for this task should be proportional to the environmental admittance. Thus, manipulation and power transfer represent conflicting requirements.

3.1.3 Advantages

3.1.3.1 Superposition of Impedances

One advantage to regarding the environment as an admittance is that one can simply superimpose the impedances coupled to that admittance, even if they are nonlinear. An intuitive justification of this property can been seen by looking at the inputs to an admittance, i.e. forces. Because the effective generalized force acting on an object is the

sum of the individual forces, the effective input to an admittance is the sum of the outputs of the attached impedances. The linearity of the individual impedances is not an issue.

This property of superposition allows one to define a number of impedances, each corresponding to one objective in a complex composite task, and use the summed impedance to accomplish the task.

3.1.3.2 Automatic Inverse Kinematics

An interesting and useful benefit of the torque based implementation of impedance control is a lack of the need for the inverse kinematic equations. All quantities are specified in Cartesian space for the calculation of the desired end-point acceleration, which converts to joint acceleration through the manipulator Jacobian (and the time derivative thereof). The advantage here is that the Jacobian can always be computed in a straightforward manner, while the kinematic equations cannot always be inverted analytically. While this is a very useful result, a number of things should be noted. First, the inverse Jacobian is needed in the standard implementation, as in all the methods discussed previously. This matrix can be singular, or even not square, in the case of redundant manipulators. Secondly, this benefit is dependent on the formulation, which depends on being able to command joint torques. Finally, the same benefit is seen in the resolved acceleration method.

3.1.3.3 Automatic Obstacle Avoidance

One example of an application of the property of the superposition of impedances is in obstacle avoidance. Hogan showed that it is possible define an impedance associated with objects in the workspace, even randomly moving objects, supposing that their instantaneous positions could be sensed. The end-effector would also have an impedance moving with it, representing its desired dynamic response. The impedance used in the calculation of end-point accelerations was then the sum of each individual impedance. The set-point trajectory given to the controller was a simple straight line, but the composite impedance function forced the manipulator to avoid the obstacles and to satisfy the required dynamic response all at once. Essentially, this is an example of incorporating the

superimposition of fictitious force fields into the calculation of desired accelerations for the manipulator.

3.1.3.4 Colinear Force and Position Management

Although it is impossible to strictly control both force and position in any one degree-of-freedom, impedance control enables a measure of control over each by controlling the relationship between the two. The departures from desired force and position are balanced according to the target impedance, allowing trade-offs. This ability is unique to impedance control, since hybrid control is fundamentally based on controlling purely position or force in orthogonal subspaces.

3.1.4 Stability

Hogan (1987, 1988) also demonstrated the stability of his proposed implementation of impedance control. He pointed out that the true action of force-feedback controllers is to change the dynamic response of the manipulator. In fact, the action of proportional-gain force feedback is to change the effective inertia as seen from the environment. As the feedback gain is increased, the effective inertia decreases. This finding has been confirmed by other researchers (see An and Hollerbach, 1987a; Volpe and Khosla, 1993), and will be further discussed in Chapter 7.

Hogan experimentally demonstrated the stability of the impedance controller using a two-link direct-drive electric manipulator. The end-effector was equipped with a stiff two degree-of-freedom force sensor, and was commanded to collide with a concrete block. The resulting motion showed very little bouncing. However, more oscillatory yet asymptotically stable results were observed when the force sensor was disconnected. This showed that the quality of the equipment was in large part responsible for the quality of Hogan's results.

Hogan also gave a theoretical proof of the stability of certain classes of impedance controllers under a very broad set of operating conditions. He defined a class of impedances which he termed "simple impedances" that are essentially generalized spring-

dashpot systems. Using a Hamiltonian approach, he demonstrated that, if the manipulator has the dynamic response of a simple impedance, and both the manipulator and the environment are stable in isolation, the dynamic interactions between the two will not cause instability. Because linearity of either system was not assumed, stability was shown to be preserved in the face of large changes in environmental dynamics. Furthermore, Hogan showed that the stability property is robust to both kinematic errors in the controller and to the presence of interface dynamics.

There are a number of caveats to the proof, however. The first is that only the stability of the endpoint is guaranteed, not that of the manipulator configuration. Secondly, the proof assumed throughout that the nominal position trajectory is held at a constant position. No claims are made of the stability of a system with a moving set-point. Lastly, there is the assumption that the manipulator can be controlled to behave as a simple impedance. Such an impedance depends only on position errors and velocities. As Hogan himself pointed out (Hogan, 1985), the manipulator can never be made to behave as a non-inertial system. Consequently, the target impedances used subsequently in this thesis are of second-order, and do not fall into the class of simple impedances.

3.1.5 Harmony of PBIC with the Fundamentals of Impedance Control

Having explored the theoretical underpinnings of impedance control, one might be lead to question whether what we have termed "position based impedance control" is really impedance control at all. Hogan stressed the distinction between impedance and admittance, and the inappropriateness of the latter for the control of constrained manipulation. However, the impedance model in the position based approach takes a force as input and produces a position to be superimposed on the specified trajectory. Is this approach therefore an implementation of admittance control? A brief departure will now be taken to demonstrate that it is not.

Consider the block diagram of the position based approach to impedance control shown in Figure 3-1. Note that this figure has been simplified for clarity. A detailed description of position based impedance control can be found in Section 3.2.1.

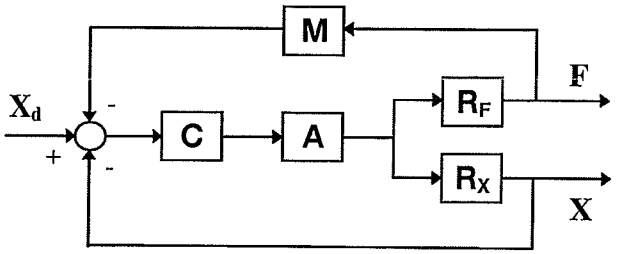


Figure 3-1: Basic position based implementation of impedance control.

The symbols used in Figure 3-1 are not constrained to being linear Laplace-space transfer functions. The block diagram is simply used to show the causal relationships between the variables, which have the following meanings:

- $\mathbf{X_d}$: the nominal position trajectory,
- $\mathbf{X}$: the actual end-effector position,
- $\mathbf{F}$: the force read by the sensor,
- $\mathbf{C}$: the position controller,
- $\mathbf{A}$: the actuators,
- $\mathbf{R_F}$,$\mathbf{R_X}$: the force and position responses of the robot-sensor-environment system, respectively, and
- $\mathbf{M}$: the model of the target impedance.

Assuming that $\mathbf{R_F}$ can be inverted, which is usually the case for a non-redundant manipulator, the control blocks can be re-arranged as follows:

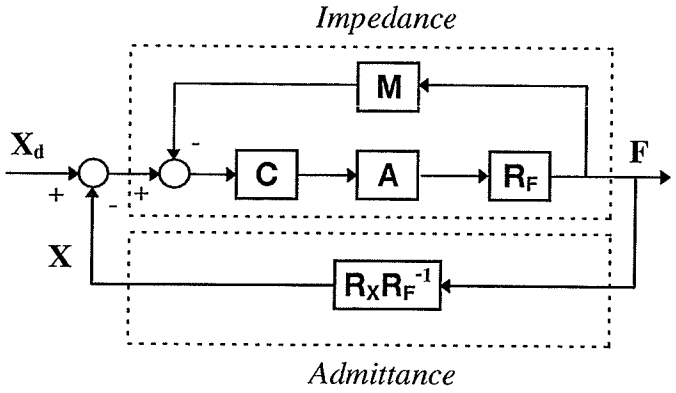


Figure 3-2: Basic position based implementation showing robot impedance modification.

Figure 3-2 clearly shows that the robot/environment and the model can be jointly considered as a single impedance. Impedance can therefore be considered to be the controlled quantity. Furthermore, the control structure cannot be arranged to enclose the model and the robot in an admittance block. However, it can be arranged to show that the model and the environment may be considered a single admittance:

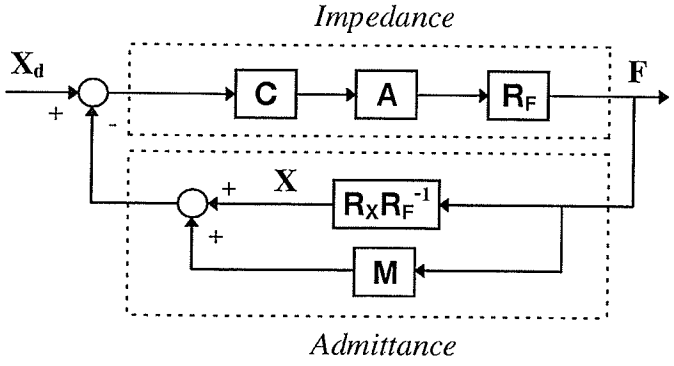


Figure 3-3: Basic position based implementation showing environmental admittance modification.

Therefore, the position based implementation may be considered as impedance control of the robot or admittance control of the environment. Since the robot is the subject of consideration here, the implementation is indeed impedance control as defined by Hogan (1985a-c). This comes as no surprise, when one considers that the first

implementation of impedance control proposed by Hogan himself was based on modifying an acceleration trajectory using force feedback. Modifying a position trajectory instead is not that different conceptually.

3.2 Linearized Analysis of Position Based Impedance Control

3.2.1 Controller Structure

Figure 3-4 shows the generalized control scheme encompassing all implementations of impedance control. The quantities **X** and **F** are vectors of the generalized position and force at the end-effector. Each block may or may not use all of the inputs shown, depending on the implementation. In general, however, the impedance controller block takes desired trajectories for both force and position and combines them with feedback information to produce a position set-point according to a target impedance. This set-point may be in the form of an actual position, as is the case in position based impedance control, or it could be a velocity or an acceleration (Hogan, 1985; Pelletier and Doyon, 1994). The set-point is passed to a position controller which controls the signal to the actuators. The position controller could be anything from a simple linear proportional controller to a computed torque feedforward scheme. The actuators produce torques according to their natures, which in turn produce forces and positions at the end-effector of the manipulator through its interaction with the environment through the force sensor.

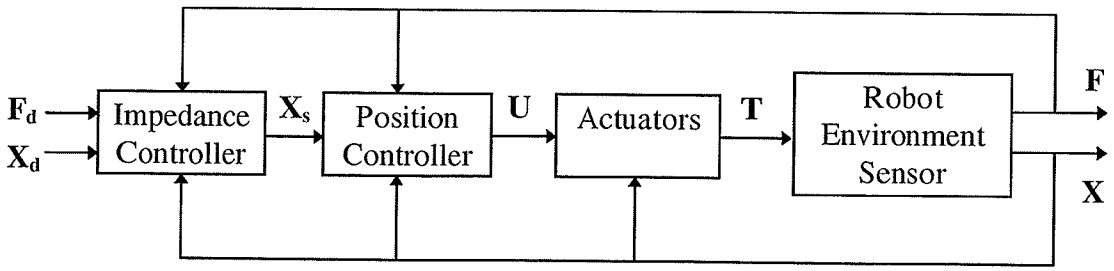


Figure 3-4: Generalized block diagram for impedance control.

Each segment of this assembly is analyzed below as it appears in a position based implementation of impedance control.

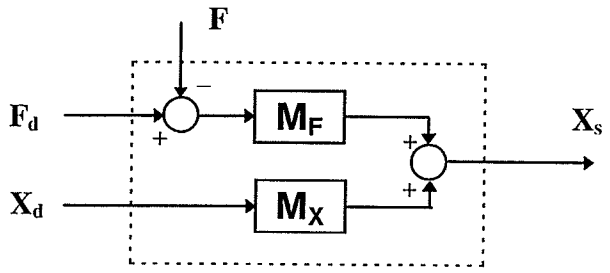


Figure 3-5: Structure of impedance controller block.

Figure 3-5 shows the general structure of the impedance controller block. In the implementation used herein, position feedback is not used in the impedance model. The one-dimensional version of the transfer functions used in the current work are second-order, as shown below. Their derivation is evident from inspection of the target impedance as illustrated in Figure 3-6.

$$\mathbf{M}_X = \frac{cs + k}{ms^2 + cs + k}$$

$$\mathbf{M}_F = \frac{1}{ms^2 + cs + k}$$

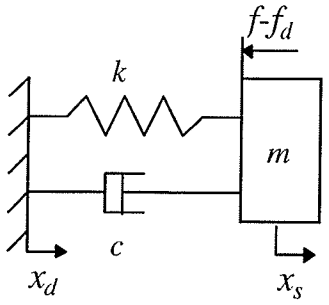


Figure 3-6: Physical model of target impedance implemented.

Note that in the model reference method introduced by Field and Stepanenko (1993), the following impedance relations were used:

$$\mathbf{M}_X = 1$$

$$\mathbf{M}_F = \frac{1}{ms^2 + cs + k}$$

Setting $\mathbf{M}_X$ equal to unity obscures the physical meaning of the desired trajectory. In the current work, the complete impedance relations are used, not just those involving force feedback.

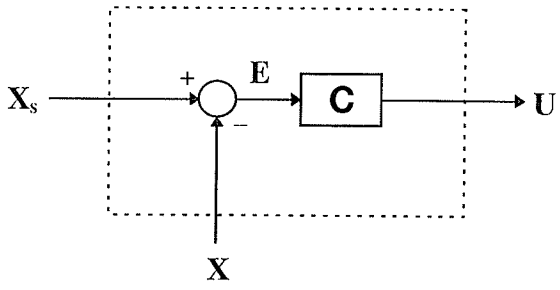


Figure 3-7: Structure of position controller block.

Figure 3-7 shows the form of the position controller considered for purposes of a linearized analysis. As already stated, the position controller need not only work on the position error signal. Some of the control strategies reviewed previously require set-point, position feedback and force feedback signals. Indeed, it will be found later that a nonlinear controller modification using joint velocities has a number of advantages over its linear counterpart.

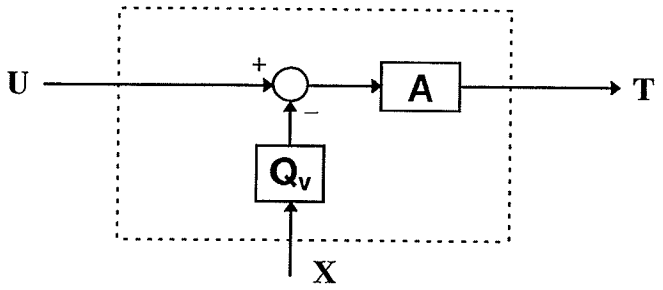


Figure 3-8: Structure of actuator block for hydraulic piston.

Assuming that the dynamics of the spool valve are fast enough to be negligible, the equations governing the actuator are as follows (D'Souza, 1988, Hanafusa et al., 1980):

$$T \propto F = A_p(P_{in} - P_{out})$$

$$\dot{P}_{inlout} = \frac{1}{C}(\pm Q_{inlout} \mp A_p \dot{x})$$

$$Q_{inlout} = K_A U \sqrt{\Delta P_{inlout}}$$

Here, T is the control torque, F is the output force of the piston, A_p is the piston area (assumed equal on both side of the piston), P_{inlout} are the pressures on either side of the piston, C is the compressibility of the hydraulic fluid, Q_{inlout} are the rates of flow of hydraulic fluid to the appropriate side, ΔP_{inlout} are the pressure differences across the spool valve for the given chamber ($\Delta P_{in} = P_{source} - P_{in}$ and $\Delta P_{out} = P_{out} - P_{return}$ for positive spool valve displacement), K_A is the orifice flow coefficient and U is the input signal. When linearized and transformed into the s-domain, these equations become:

$$\mathbf{A} = \frac{K_x A_p}{\frac{1}{2} C s + K_\pi}$$

$$\mathbf{Q_V} = \frac{A_p}{K_x} s$$

$$K_x = \frac{\partial}{\partial U}(K_A U \sqrt{\Delta P_{inlout}}) = K_A \sqrt{\Delta P_{inlout}}$$

$$K_\pi = \frac{\partial}{\partial(\Delta P)}(K_A U \sqrt{\Delta P_{inlout}}) = \frac{K_A U}{2\sqrt{\Delta P_{inlout}}}$$

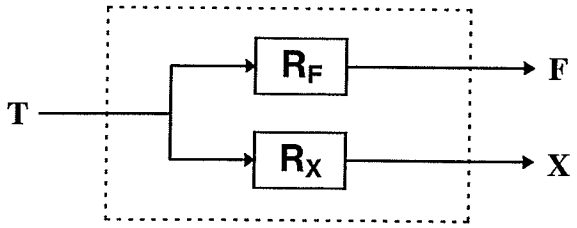


Figure 3-9: Structure of robot-environment-sensor block.

Eppinger and Seering (1986) showed that at least the first resonant mode of the manipulator must be included in any model of its dynamics in order to capture possible instability in force control. They suggested a sixth order model of the complete robot-sensor-environment dynamics, but showed that a fourth order model can also encapsulate unstable behaviour. Volpe and Khosla (1993a-b) also showed how a fourth order model could be used to analyze the dynamics of the coupled system. They suggested that the dynamics of the sensor and those of the manipulator can be lumped together if they are similar. The forces at the end-effector are equal to the compression of the sensor multiplied by its stiffness ($f = k_2(x\text{-}x_{env})$, see Figure 3-10). This model of the coupled system is shown in Figure 3-10, and results in the transfer functions given below:

$$\mathbf{R_X} = \frac{m_B s^2 + (c_s + c_e)s + (k_s + k_e)}{\Delta_R(s)}$$

$$\mathbf{R_F} = \frac{k_s(m_B s^2 + c_e s + k_e)}{\Delta_R(s)}$$

$$\Delta_R(s) = \left(m_B s^2 + (c_s + c_e)s + (k_s + k_e)\right)\left(m_A s^2 + c_r s + k_r\right) + \left(m_B s^2 + c_e s + k_e\right)\left(c_s s + k_s\right)$$

In the event that contact with the environment is lost, m_B, c_e and k_e become essentially zero, and the above equations simplify to:

$$\mathbf{R}_X = \frac{1}{m_A s^2 + c_r s + k_r}$$
$$\mathbf{R}_F = 0$$

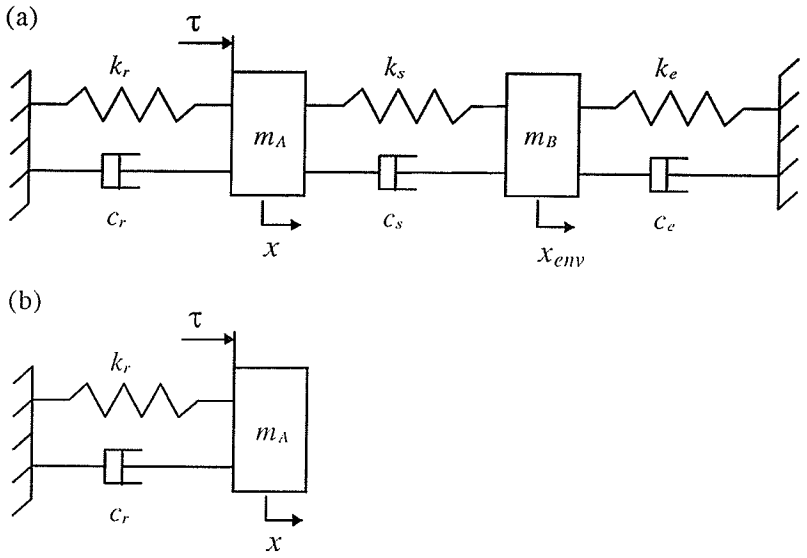


Figure 3-10: Model of robot-environment-sensor dynamics, (a) in contact with the environment, (b) in free motion (Volpe and Khosla, 1993b).

Given the structure of the various components of the complete system, linearized transfer functions between inputs X_d and F_d, and outputs X and F can be derived. Considering the actuator torque, T, a small number of steps lead to the results below:

$$\begin{aligned} T &= \mathbf{A}(U - \mathbf{Q}_v X) \\ &= \mathbf{A}(\mathbf{C}(X_s - X) - \mathbf{Q}_v X) \\ &= \mathbf{A}(\mathbf{C}(\mathbf{M}_X X_d + \mathbf{M}_F(F_d - F) - X) - \mathbf{Q}_v X) \\ &= \mathbf{A}(\mathbf{C}(\mathbf{M}_X X_d + \mathbf{M}_F(F_d - \mathbf{R}_F T) - \mathbf{R}_X T) - \mathbf{Q}_v \mathbf{R}_X T) \\ T &= \frac{\mathbf{AC}(\mathbf{M}_X X_d + \mathbf{M}_F F_d)}{1 + \mathbf{R}_F \mathbf{ACM}_F + \mathbf{R}_X \mathbf{A}(\mathbf{C} + \mathbf{Q}_v)} \end{aligned}$$

Using the above result and the transfer functions obtained from the fourth-order model of the couple mechanical system, the following overall transfer functions are obtained:

$$X = \frac{\mathbf{R_X AC(M_X}\, X_d + \mathbf{M_F}\, F_d)}{1 + \mathbf{R_F ACM_F} + \mathbf{R_X A(C} + \mathbf{Q_v)}}$$

$$F = \frac{\mathbf{R_F AC(M_X}\, X_d + \mathbf{M_F}\, F_d)}{1 + \mathbf{R_F ACM_F} + \mathbf{R_X A(C} + \mathbf{Q_v)}}$$

These derivations and their results will be used subsequently for analysis.

Note that above transfer functions are easily extended into all the degrees of freedom of the end-effector by replacing the scalar impedance parameters with their matrix equivalents. For example, the multi-degree-of-freedom model transfer functions are:

$$\mathbf{M_X} = \left(\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K}\right)^{-1}\left(\mathbf{C}s + \mathbf{K}\right)$$

$$\mathbf{M_F} = \left(\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K}\right)^{-1}$$

where **M**, **C**, and **K** and the matrix equivalents of the impedance model mass, damping, and stiffness, respectively. The transfer functions for the remaining blocks in Figure 3-4 can be generalized in a like manner.

3.3 Summary

Impedance control was presented as a general approach to manipulation. The theoretical underpinnings of impedance control were stated. A position based formulation of impedance control was developed for implementation on an industrial hydraulic manipulator. This formulation was shown to be consistent with the fundamentals of impedance control. The linearized dynamics of the composite controller-robot-sensor-environment system were derived for later use.

4. Development of a Nonlinear PI Controller

4.1 The Need for a Robust and Accurate Position Controller

The success of the position based impedance control scheme relies on the efficacy of the position controller. When in contact with a stiff environment, the force at the end-effector will be very dependent on small changes in position. Furthermore, as will be seen in Chapter 7, the smooth operation of the impedance control system depends on its ability to quickly and accurately follow small changes in position set-point.

Designing a quick, accurate and robust position controller for a hydraulic system is a challenging problem. The controller must contend with many nonlinearities and nonidealities in the system -- nonlinear dynamics, actuator deadband, and stiction are all present. It will be seen in this chapter that a simple PI controller may not be able to meet these requirements. In an attempt to deal with these problems, four candidate controllers were tested.

4.2 Experimental Setup

The Unimate MKII-2000 robot was used for all the experiments described in this work. It was an industrial hydraulic manipulator that had seen 20 years of service in the railroad industry. It had a spherical geometry, i.e. the first three degrees of freedom were left/right, up/down, and in/out, in that order. Its hydraulic actuators were supplied by a constant-pressure source operating at 7000±500kPa. The spool valves had a parabolic orifice area to spool displacement function and 6% deadband.

Position feedback was obtained using digital encoders with resolutions of 0.028°, 0.028°, and 0.105mm on links 1, 2, and 3, respectively. These readings were taken by a 486 DX2/66 based PC, which also performed all control actions. The control signals saturated at 360mA.

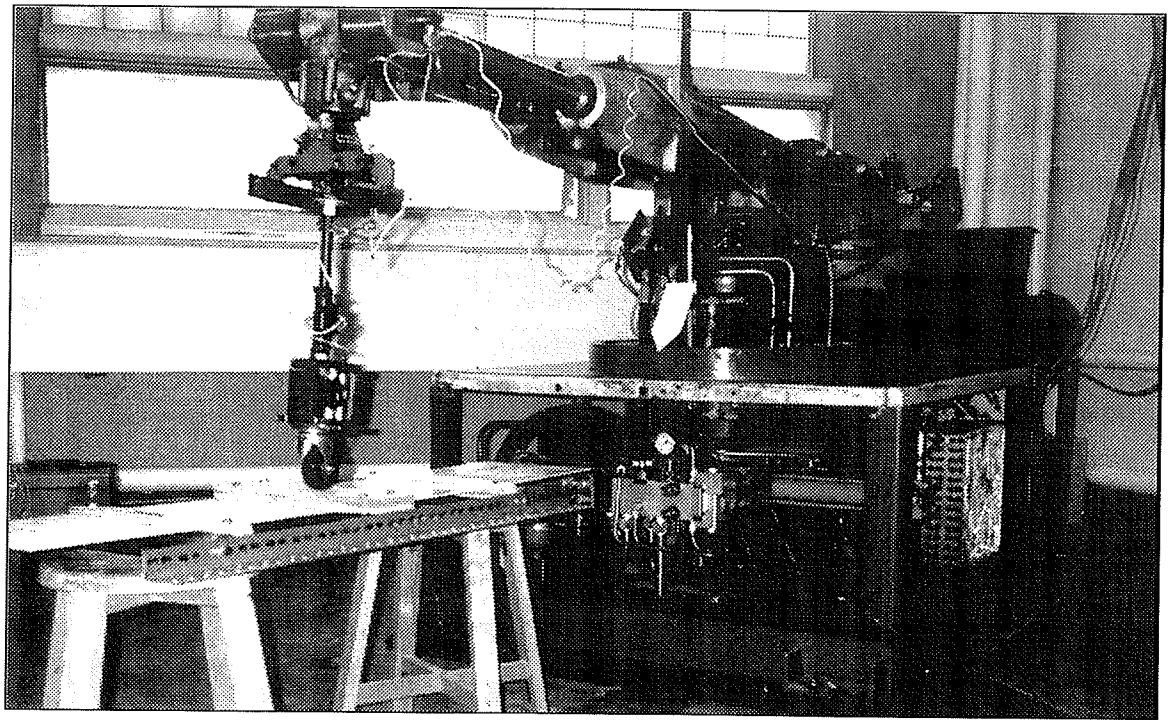


Figure 4-1: The Unimate MKII-2000.

The tests were performed using only the second link of the Unimate MKII. The average controller sampling time was 4.5 milliseconds. Joint velocities were calculated

using a 20-point linear regression on the joint encoder readings (see Appendix B for details).

Five sets of position trajectories were used as test patterns for each controller. Two sets of multiple step patterns and two sets of multiple ramp trajectories with parabolic (smooth) accelerations were used. Together, these four patterns tested the controller's ability to track changes in position and velocity of over two orders of magnitude. The final test was possibly the most demanding: a 1Hz sine wave of diminishing amplitude. This particular test was chosen to characterize the controller's dynamic performance at various magnitudes of input. The trajectory was also similar to the type of trajectory that would have to be followed in an underdamped impedance control application.

4.3 Linear PI Control

A simple PI control law was the first to be tested. Actually, strict linear PI control was modified by setting position error signals below the encoder resolution in magnitude to zero. This modification was necessary to prevent random walk in the integral term, which was observed to induce limit cycles even when no input was given to the system.

$$u(t) = K_P e(t) + K_I I(t)$$

$$e(t) = q_s(t) - q(t)$$

$$I(t) = I(t - \Delta t) + e(t)\Delta t$$

The control signal, $u(t)$, is the weighted sum of $e(t)$, the error between the joint position, $q(t)$, and the joint set-point, $q_s(t)$. $I(t)$ is the time integral of $e(t)$. Using controller gains of K_P = 215mA·deg^{-1} and K_I = 115mA·deg^{-1}s^{-1}, this method proved very effective for most of the tests. Trajectory tracking was good for the large ramp (Figure 4-2) and small step (Figure 4-5) patterns. However, integral wind-up caused expectedly large overshoots in response to large step inputs (Figure 4-3). The effects of stiction and actuator deadband were most apparent in the complete failure to track the sine wave pattern (Figure 4-6). These factors also had visible effects in both ramp tests. Delays in motion of up to 2 seconds, due to stiction and actuator deadband, were observed (Figure 4-4). Ramp tests also showed overshoot in the manipulator response.

The remaining methods examined all attempted to deal with the problems of stiction and wind-up. These methods employ nonlinear techniques to increase sensitivity while preserving stability.

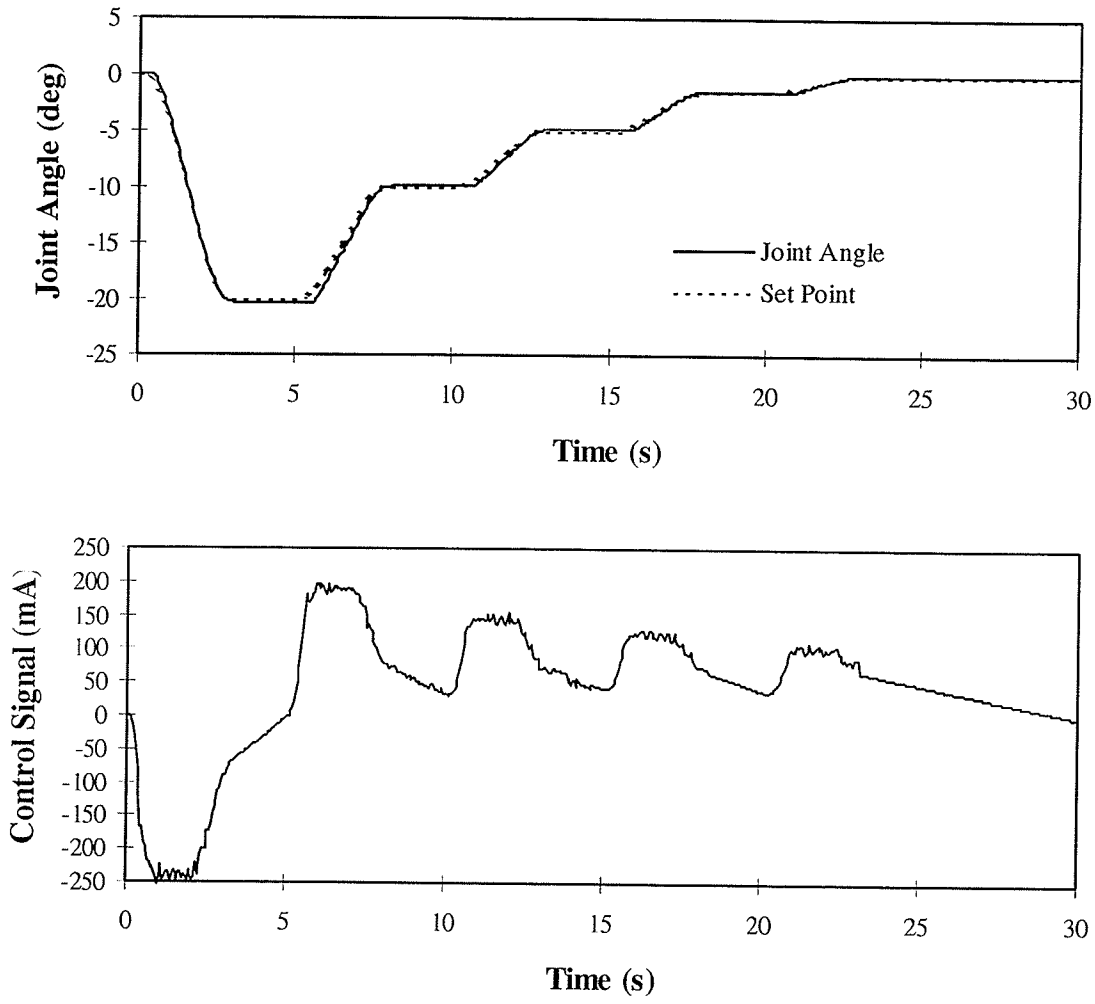


Figure 4-2: PI control, large ramp pattern (K_P=215mA·deg^{-1}, K_I=115mA·deg^{-1}s^{-1}).

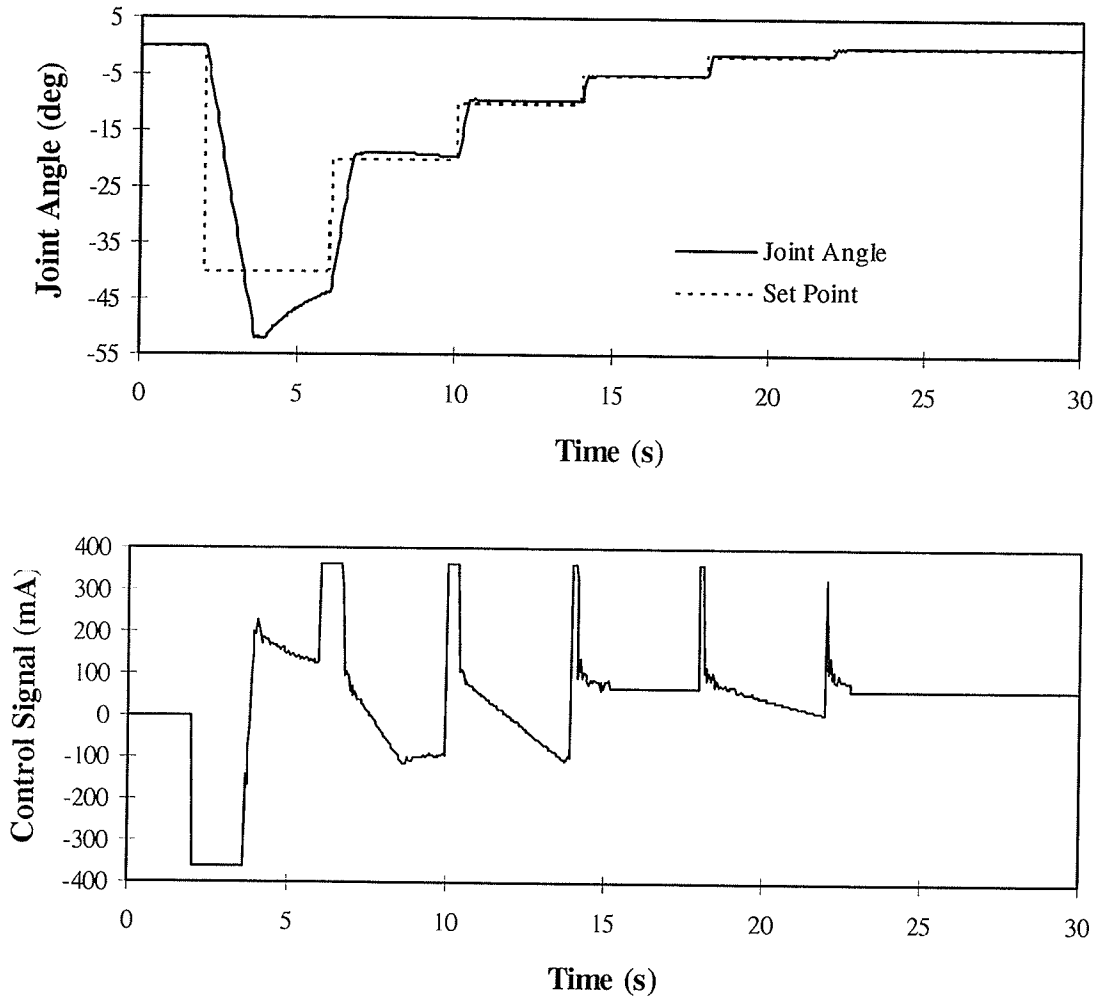


Figure 4-3: PI control, large step pattern (K_P=215mA·deg^{-1}, K_I=115mA·deg^{-1}s^{-1}).

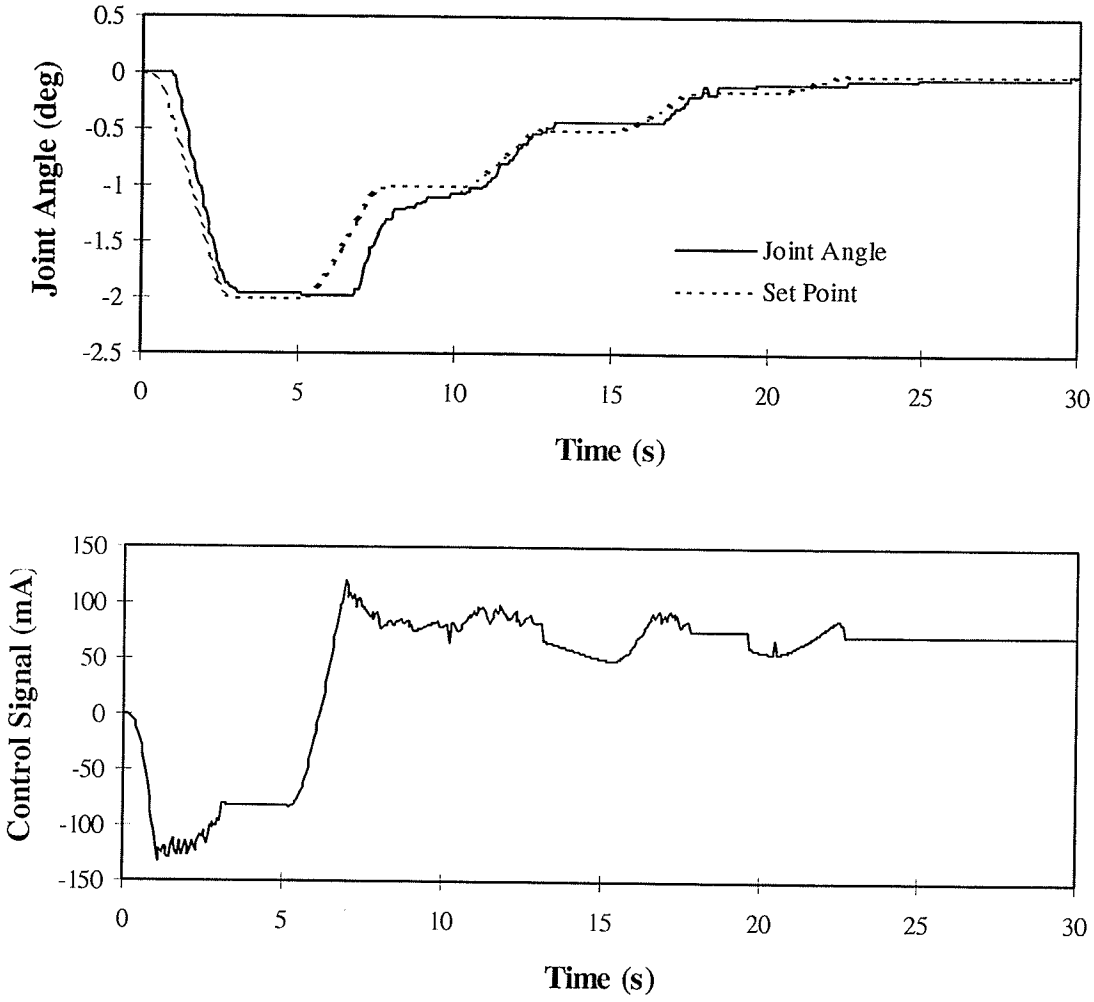


Figure 4-4: PI control, small ramp pattern (K_P=215mA·deg^{-1}, K_I=115mA·deg^{-1}s^{-1}).

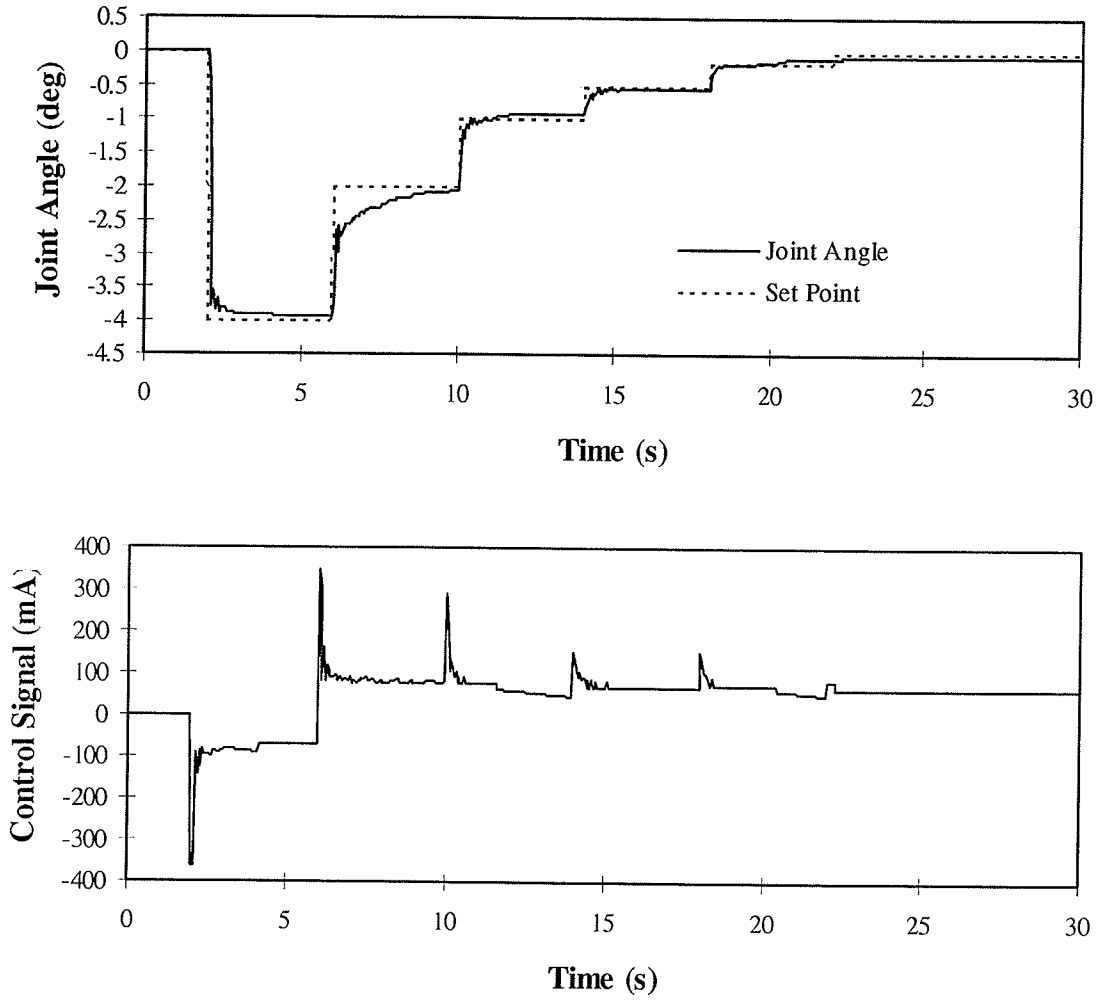


Figure 4-5: PI control, small step pattern (K_P=215mA·deg^{-1}, K_I=115mA·deg^{-1}s^{-1}).

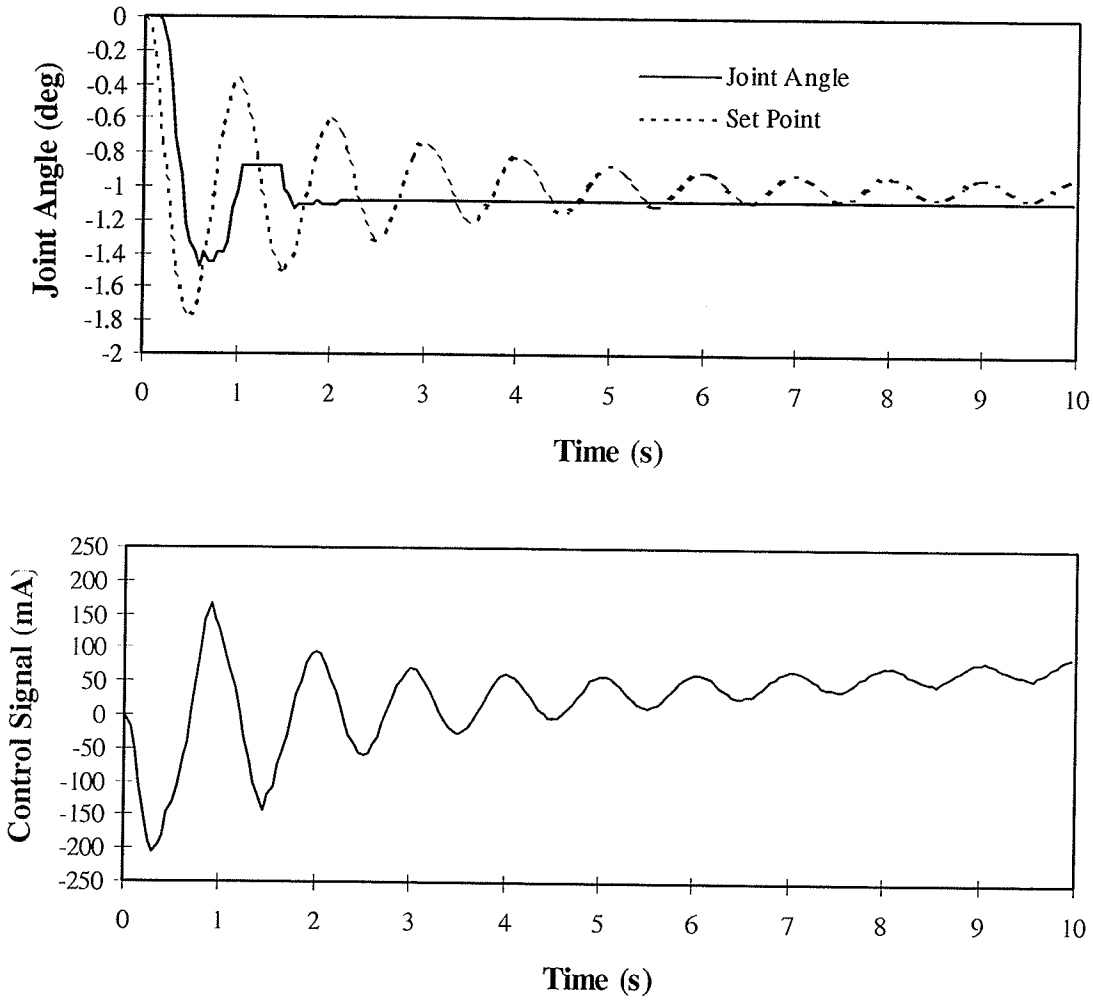


Figure 4-6: **PI control, decreasing sine wave pattern (K_P=215mA·deg^{-1}, K_I=115mA·deg^{-1}s^{-1}).**

4.4 Modified Morse's Method

Morse's patented controller uses a proportional plus nonlinear integral method. The implementation is digital in character. The algorithm differs from linear PI control in the calculation of the error integral I, as shown below:

$$I(t) = (I(t - \Delta t) + e(t)\Delta t)\frac{\alpha}{\alpha + \dot{q}^2(t)}$$

Here, α is an arbitrary controller gain, q is the controlled joint angle, e is the joint position error, t is the time and Δt is the time interval. The nonlinear factor is unity at low velocities, and goes to zero at high velocities. The intent is to reduce the effects of stiction and eliminate overshoot. The effects of stiction should be reduced because the nonlinear factor is less than unity for any non-zero joint velocity, allowing the use of higher integral gains and increasing the control action at low speeds. Since the entire accumulated integral is iteratively multiplied by this factor, the integral resets itself at high velocities, reducing overshoot.

Morse's method works by resetting the integral at high speeds. This approach works well for step changes in set-point. However, if high-speed trajectory tracking is desired, Morse's method is no better than straight P-action. Therefore, the following modification to Morse's integral calculation method was made:

$$I(t) = (I(t - \Delta t) + e(t)\Delta t)\frac{\alpha(\dot{q})}{\alpha(\dot{q}) + \dot{e}^2(t)}$$

Using the velocity error in the nonlinear factor instead of the velocity itself has two advantages. The first is that the integral will only reset in situations where integral wind-up is a possibility, i.e. when the set-point is rapidly changing. The second advantage is that, for a step change in the set-point, the modified method is indistinguishable from the original.

The disadvantage to this modification is that the anti-stiction properties of the original Morse controller are slightly degraded. When the set-point is not being followed due to the effects of static friction, the joint velocity is zero but the velocity error is large. The original Morse method's nonlinear factor is unity during this period, providing full controller action, whereas the modified nonlinear factor is small when stiction occurs, and is therefore not as effective as the original factor. However, experimental results showed that the higher gains allowed by the incorporation of the nonlinear factor still provided improved response time over linear PI control.

One other modification was made to the controller. To assist in fine movements, the cutoff velocity parameter α in the modified controller was linked to the set-point velocity, $\dot{q}_s$. The controller was tuned using a certain maximum velocity, $\dot{q}_{\max}$. The value of α_{max} thereby obtained was then used in the calculation of α as follows:

$$\alpha = \alpha_{\max}\left(0.001 + \frac{|\dot{q}_s|}{\dot{q}_{\max}}\right)$$

This modification was found to make little difference in the overall behaviour of the manipulator. However, it did reduce the overshoot observed for very small changes in position. This reaction was observed because a small α brought the integral to zero very quickly after any movement. Effectively, the controller was made to "nudge" the manipulator. This nudging action is especially apparent at approximately the 4 second mark in Figure 4-7, and on either side of the 2 second mark in Figure 4-9.

The modified Morse method shows definite improvements over the linear PI controller. The nonlinear factor greatly improved the large step response; compare Figure 4-2 with Figure 4-8. The nonlinear factor also allowed a ten-fold increase in the integral gain (see Appendix C), improving tracking in ramp tests (Figure 4-7 and Figure 4-9). As predicted, problems with sticking remain. Both these points are illustrated in Figure 4-11, which shows greatly improved sine wave tracking at large amplitudes, but renewed sticking for small commanded displacements. Finally, overshoot remained in ramp and sine wave tracking. The next refinement will address these issues.

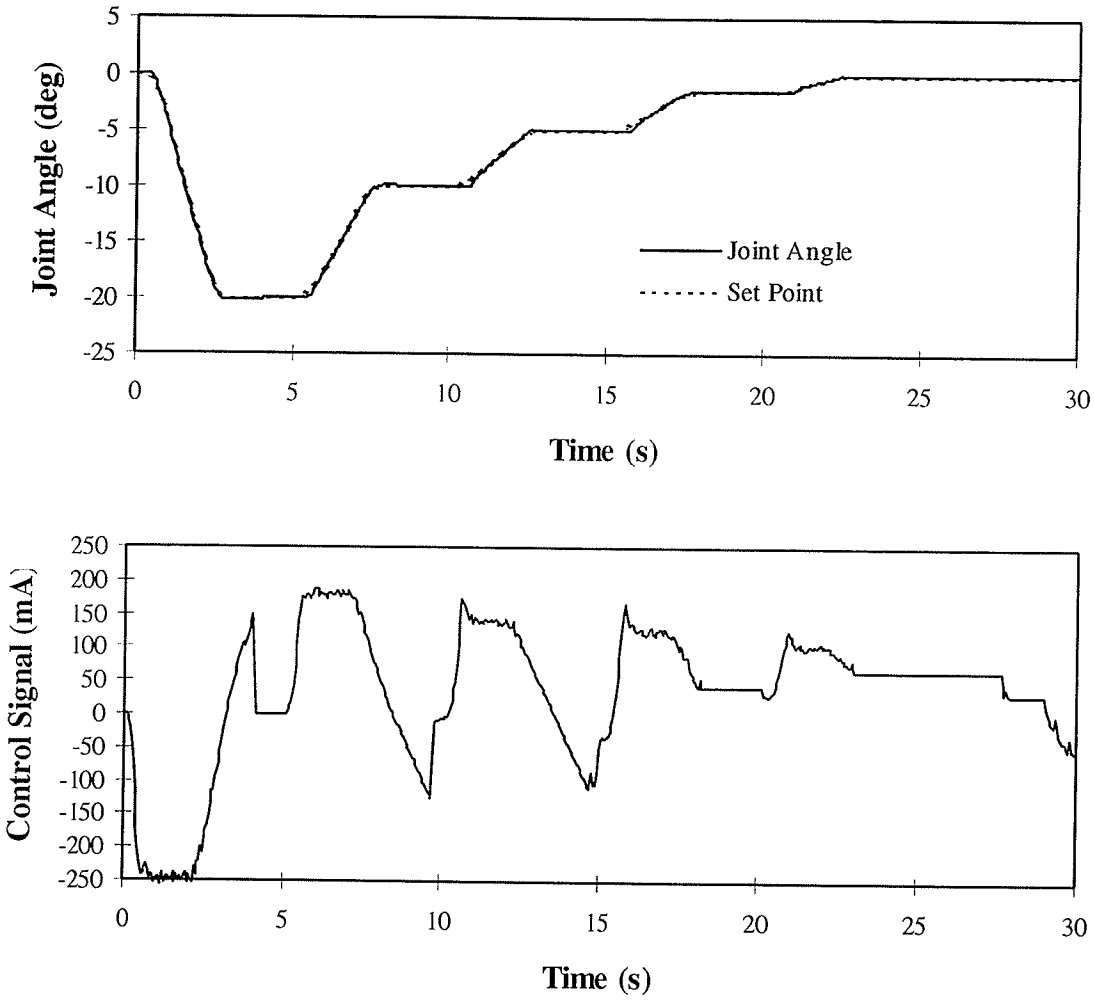


Figure 4-7: **Modified Morse method, large ramp pattern (K_P=215mA·deg^{-1}, K_I=1151mA·deg^{-1}s^{-1}, α_{max}=8000deg^{2}s^2, $\dot{q}_{max}$=10deg/s).**

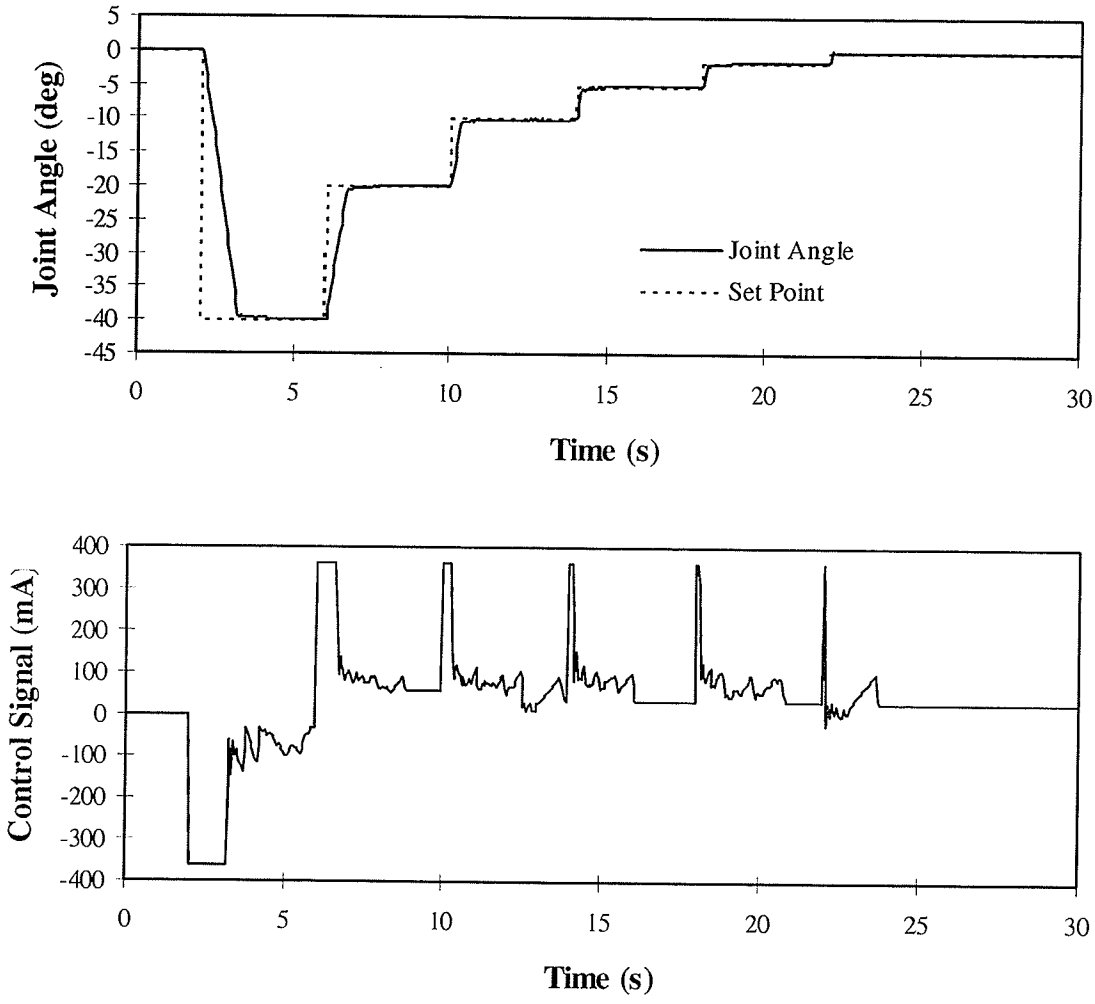


Figure 4-8: **Modified Morse method, large step pattern (K_P=215mA·deg^{-1}, K_I=1151mA·deg^{-1}s^{-1}, α_{max}=8000deg^{2}s^2, $\dot{q}_{max}$=10deg/s).**

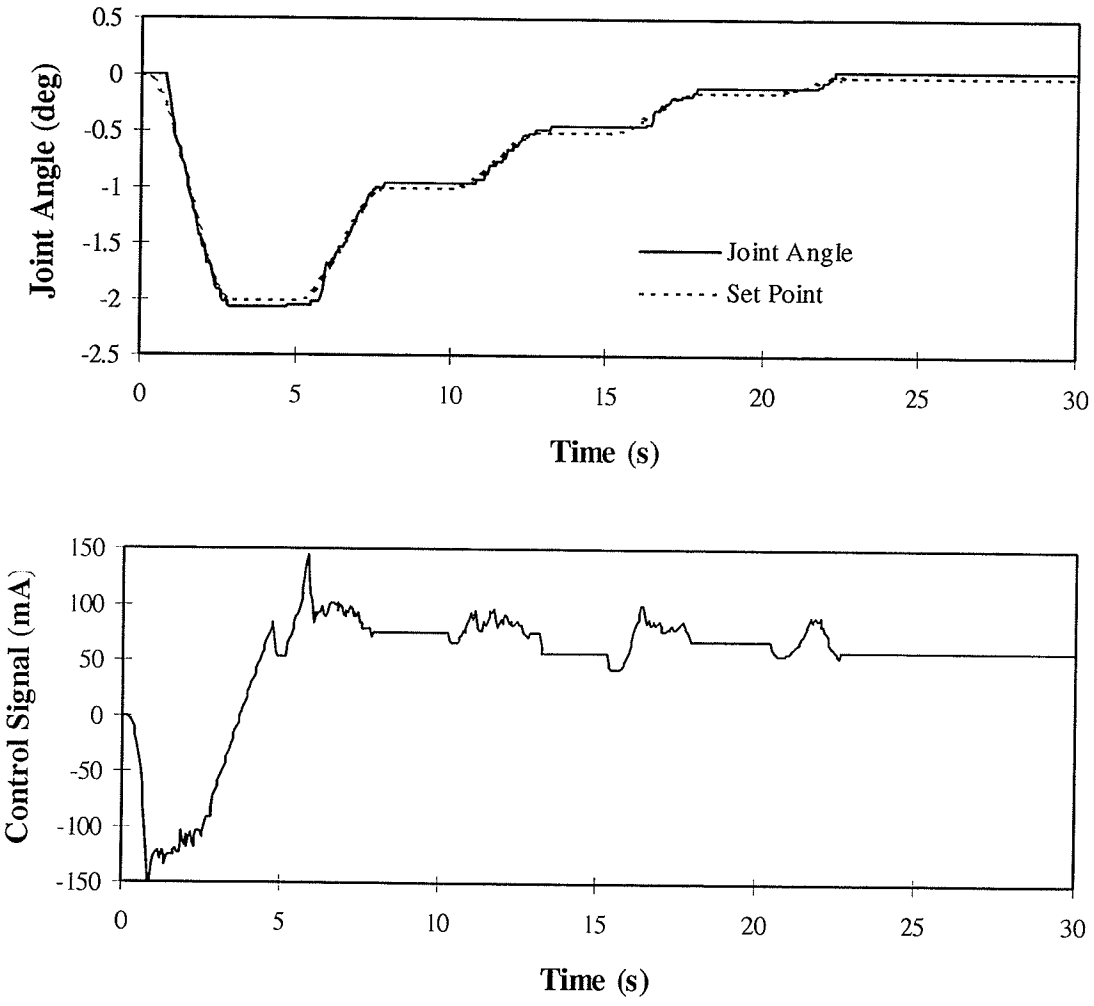


Figure 4-9: **Modified Morse method, small ramp pattern (K_P=215mA·deg^{-1}, K_I=1151mA·deg^{-1}s^{-1}, α_{max}=8000deg^{2}s^2, $\dot{q}_{max}$=10deg/s).**

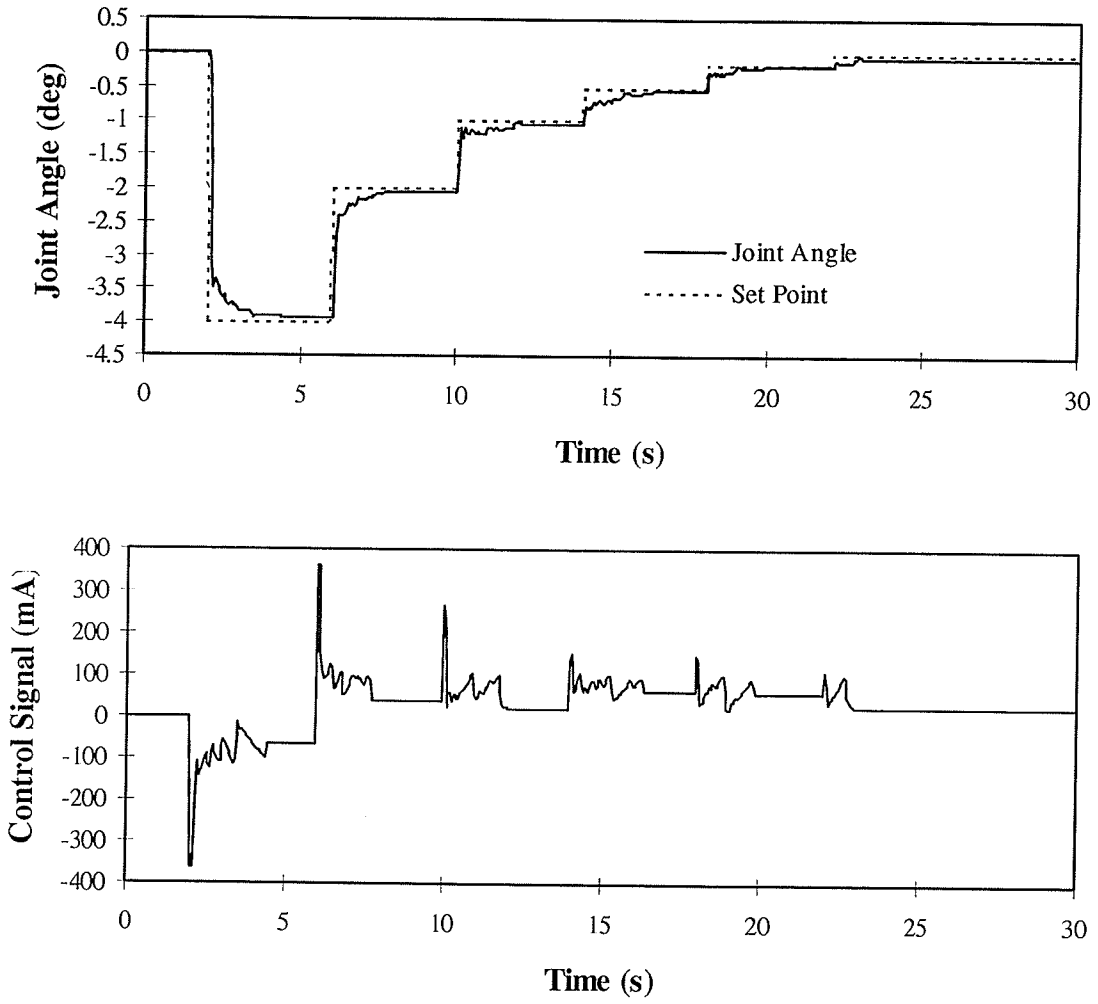


Figure 4-10: Modified Morse method, small step pattern (K_P=215mA·deg^{-1}, K_I=1151mA·deg^{-1}s^{-1}, α_{max}=8000deg^{2}s^2, $\dot{q}_{max}$=10deg/s).

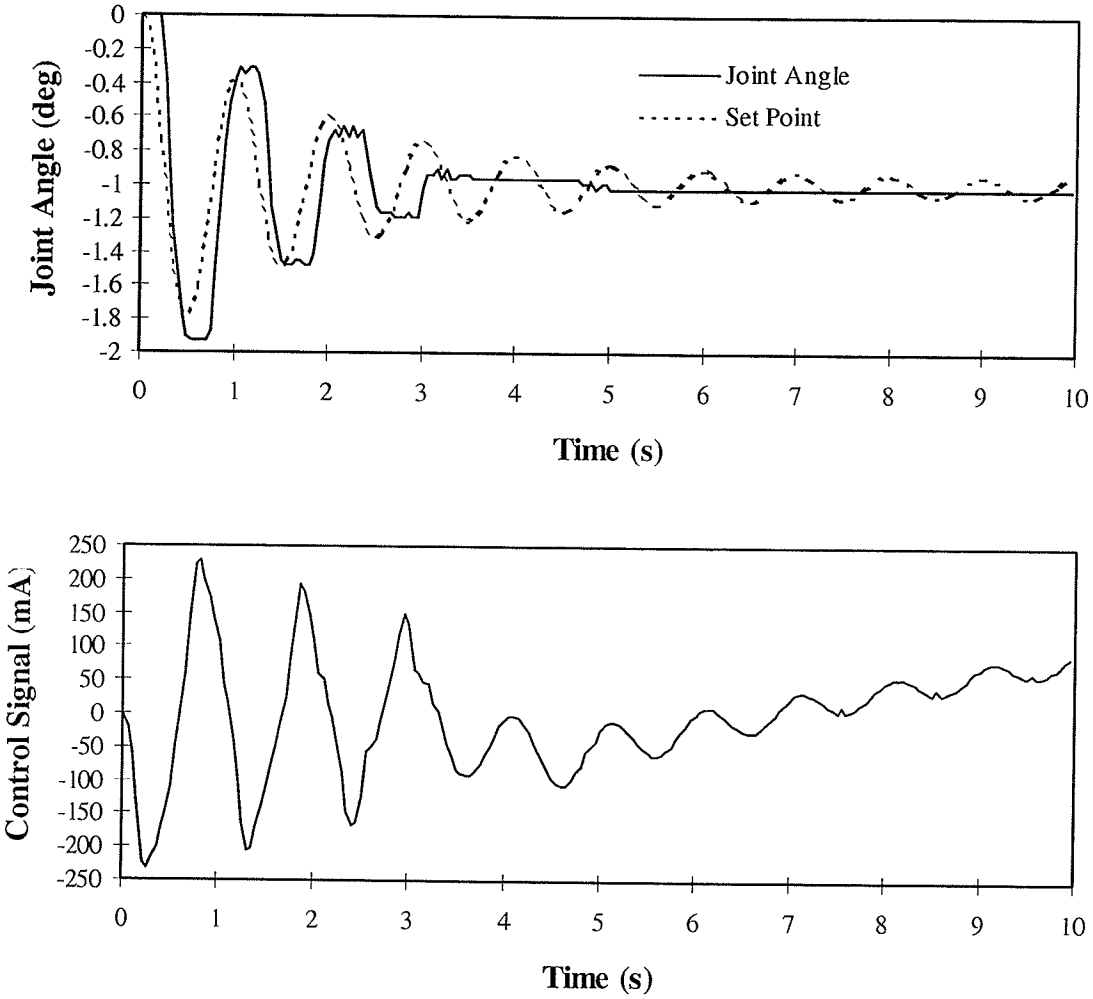


Figure 4-11: Modified Morse method, decreasing sine wave test (K_P=215mA·deg^{-1}, K_I=1151mA·deg^{-1}s^{-1}, α_{max}=8000deg^{2}s^2, $\dot{q}_{max}$=10deg/s).

4.5 Velocity Error Triggered Integral Augmentation

A number of techniques were reviewed in an attempt to negate the effects of stiction. These methods are discussed below, but, in the interests of brevity, the experimental results are shown for the most successful method only, the velocity error triggered integral augmentation method.

4.5.1 Deadband Compensation

The aim of this method was to eliminate the effects of overlap in the spool valve. Based on the observation that the control signal always had to exceed a threshold of ~88mA in absolute value before motion occurred (see, for example, Figure 4-7), the signal augmentation function $u_c(t)$ was defined as shown in Figure 4-12. This signal was added to the control signal $u(t)$, which was calculated by the Modified Morse method.

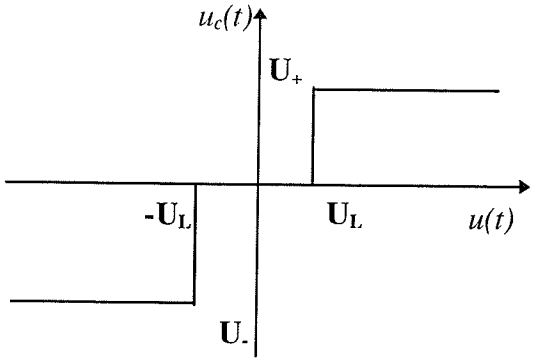


Figure 4-12: Deadband compensation.

Using this function (with U_L = 7mA, U_+ = $-U_-$ = 88mA) on the control signal obtained using Morse's method gave excellent trajectory tracking for large displacements. However, the method was less successful when dealing with small displacements. Small control signals caused rapid bang-bang action in the augmentation signal, inducing oscillations in the system. When dealing with an environment of any reasonable stiffness, the commanded motions will be small ones, making this method of deadband compensation unsuitable for application to impedance control.

4.5.2 Deadband Compensation with Hysteresis

A variant of deadband compensation that includes hysteresis was implemented in an attempt to retain the favourable large-displacement response for small motions. Under this modification, the added control signal changes only after the underlying control signal had definitely switched signs. The added signal is shown in Figure 4-13.

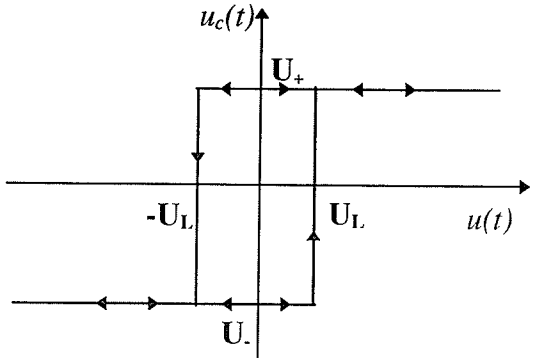


Figure 4-13: Deadband compensation with hysteresis.

The same parameters were used here as previously: U_L = 7mA, $U_+ = -U_-$ = 88mA. Inclusion of hysteresis proved to extend the trajectory-tracking abilities of the controller to small motions. The rapid signal switching observed was eliminated. However, while this modification successfully extended the controller's trajectory-tracking abilities, it adversely affected its static regulation abilities. When tested with a simple step input, the controller brought the manipulator to the desired position very quickly but was unable to hold it steady since the added signal had no "off" position. To regain static stability, either the signal threshold U_L or the added signal U_+/U_- must be decreased. However, either of these actions would result in an increased time required for the control signal to reach the levels required to compensate for the deadband. Therefore, increasing static stability would directly affect the controllers ability to reduce sticking, which is precisely the reason for its introduction. Evidently, a better method of stiction compensation must be found.

4.5.3 Nonlinear Velocity Feedback

Another avenue of stiction compensation explored was that of using a nonlinear velocity feedback signal. The motivation for this approach comes from the observation that the largest error signal during an occurrence of stiction is the velocity error.

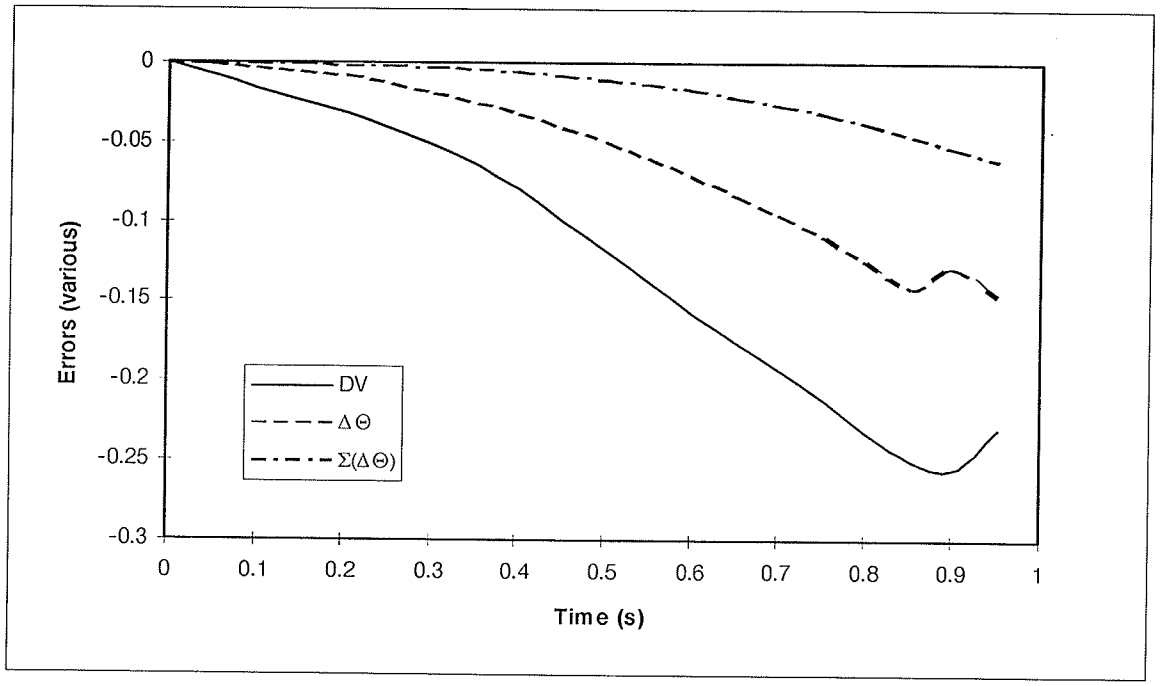


Figure 4-14: Velocity error, position error, and error integral signals in stiction (static friction broken at t = 0.9s).

Figure 4-14 is taken from the first second of the trial shown in Figure 4-9. The labels $\Delta\Theta$, DV, and $\Sigma(\Delta\Theta)$ refer to the position error (in degrees), its time derivative, and its time integral, repsectively. Inspection of Figure 4-14 shows that integral based methods of stiction compensation rely on the smallest signal. The velocity error provides a much stronger signal. However, a nonlinear factor must be incorporated in any use of this signal to avoid the damping action of derivative feedback. The result of this nonlinear filtering is an estimate of the velocity error caused by stiction:

$$\dot{e}_{sticiton} = (\dot{q}_s - \dot{q})\frac{\dot{q}_s^2}{\dot{q}_s^2 + \beta\dot{q}^2}$$

$\dot{e}_{stiction}$ is the estimate, $\dot{q}$ and $\dot{q}_s$ are the actual and set-point joint velocities, and β is a parameter increasing selectivity. The nonlinear factor is greatest when $\dot{q}$ is zero, which is when stiction occurs. The terms involving $\dot{q}_s$ were incorporated so that the estimate would be zero when the set-point was unchanging. Otherwise, the velocity error estimate in feedback would cause extra damping when the manipulator was slowly approaching its set-point. In effect, it would work against any integral terms attempting to eliminate steady-state error. The modification prevents this scenario from occurring.

$\dot{e}_{stiction}$ is shown in Figure 4-15. The estimate is appreciable only in the portion of the velocity plane where stiction occurs: along the line $\dot{q} = 0$.

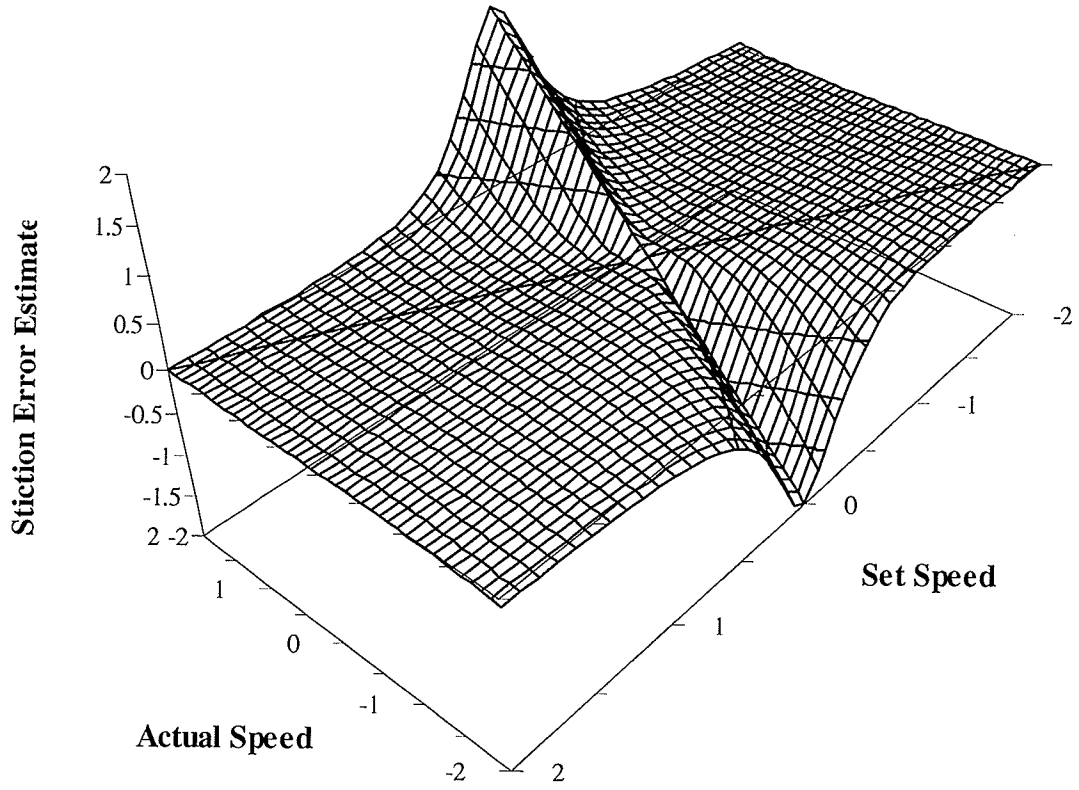


Figure 4-15: Region of operation of proposed stiction correction signal (β=50).

The problem with using $\dot{e}_{stiction}$ in direct feedback, i.e. $U = K_P e + K_I I + K_V \dot{e}_{stiction}$, is that the estimate, and therefore the control signal, will drop to near zero with any motion. Motion will consequently be intermittent. Therefore, direct feedback is infeasible. However, it will now be shown below how nonlinear velocity feedback can be successfully used in conjunction with the previously presented techniques to practically eliminate sticking.

4.5.4 Velocity Error Triggered Integral Augmentation (VETIA)

The method presented here exploits the strengths of three of the previously reviewed methods: the modified Morse's method, deadband compensation, and nonlinear velocity feedback. The superior tracking and step response of the modified Morse's method is combined with the stiction elimination seen in the most ideal trials using deadband compensation. The key to the method is the nonlinear velocity feedback signal discussed previously.

Velocity error triggered integral augmentation uses nonlinear velocity feedback as a switching signal. Once the experimentally determined threshold $\dot{e}_{\min}$ is met, deadband compensation is employed. However, because the modified Morse control signal is no longer needed for the activation criterion, the extra signal can be directly added to the integral term in the modified Morse controller. The algorithm used is shown below:

$$u(t) = K_p e(t) + K_I I(t)$$

$$I(t) = \begin{cases} \dfrac{-U_{lower} - K_p e(t)}{K_I} & \dot{e}_{stiction} < -\dot{e}_{min} \;\&\; u(t) > -U_{lower} \\ (I(t-\Delta t) + e(t)\Delta t)\dfrac{\alpha(\dot{q})}{\alpha(\dot{q}) + \dot{e}^2(t)} & |\dot{e}_{stiction}| \le \dot{e}_{min} \\ \dfrac{U_{upper} - K_p e(t)}{K_I} & \dot{e}_{stiction} > \dot{e}_{min} \;\&\; u(t) < U_{upper} \end{cases}$$

Note that the integral augmentation outght to occur only when the velocity error was first detected, so that the integral does not get perpetually reset if motions do not instantly ensue (see Appendix C for a formulation which implements this feature).

The parameters used for testing this controller are given in Appendix C. Inspection of the experimental results shows the superior trajectory tracking abilities obtained by augmenting the modified Morse controller with the VETIA method. Figure 4-16 & Figure 4-20 especially show how the control action was modified to improve response time. The method is robust, proving itself on all tests. In no test does the proposed method show any detrimental side effects. The control signal is merely speeded up in reaching the necessary levels for actuation. Therefore, while the response time is quickened, the ensuing control signal action is unchanged.

The only undesirable behaviour remaining in the new position controller is the slight overshoot seen in both PI control and the modified Morse method. In a force control application, this overshoot could be very detrimental to performance. It will be dealt with by a final modification to the position controller.

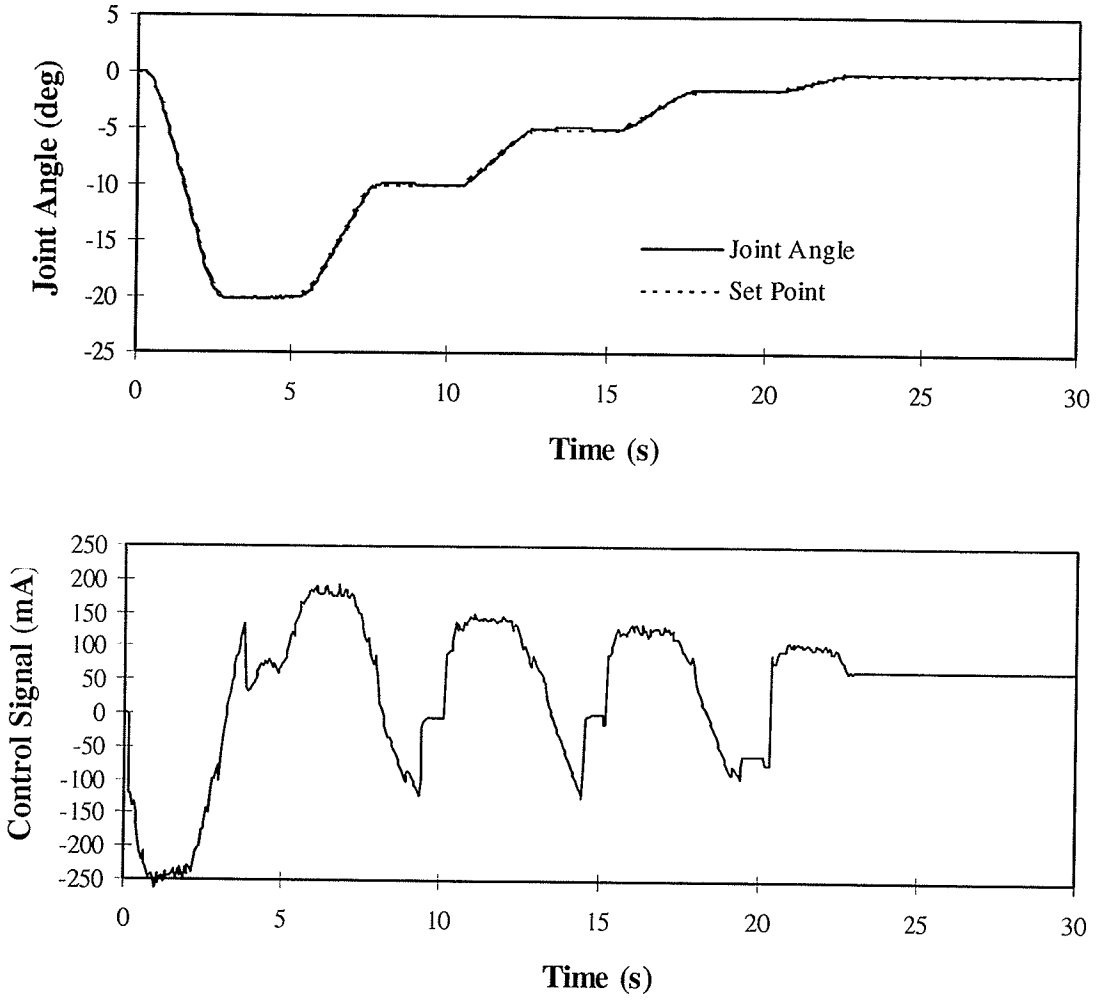


Figure 4-16: VETIA, large ramp pattern ($\dot{e}_{min}$ =0.15deg/s, U_{lower}=-88mA, U_{upper}=114mA).

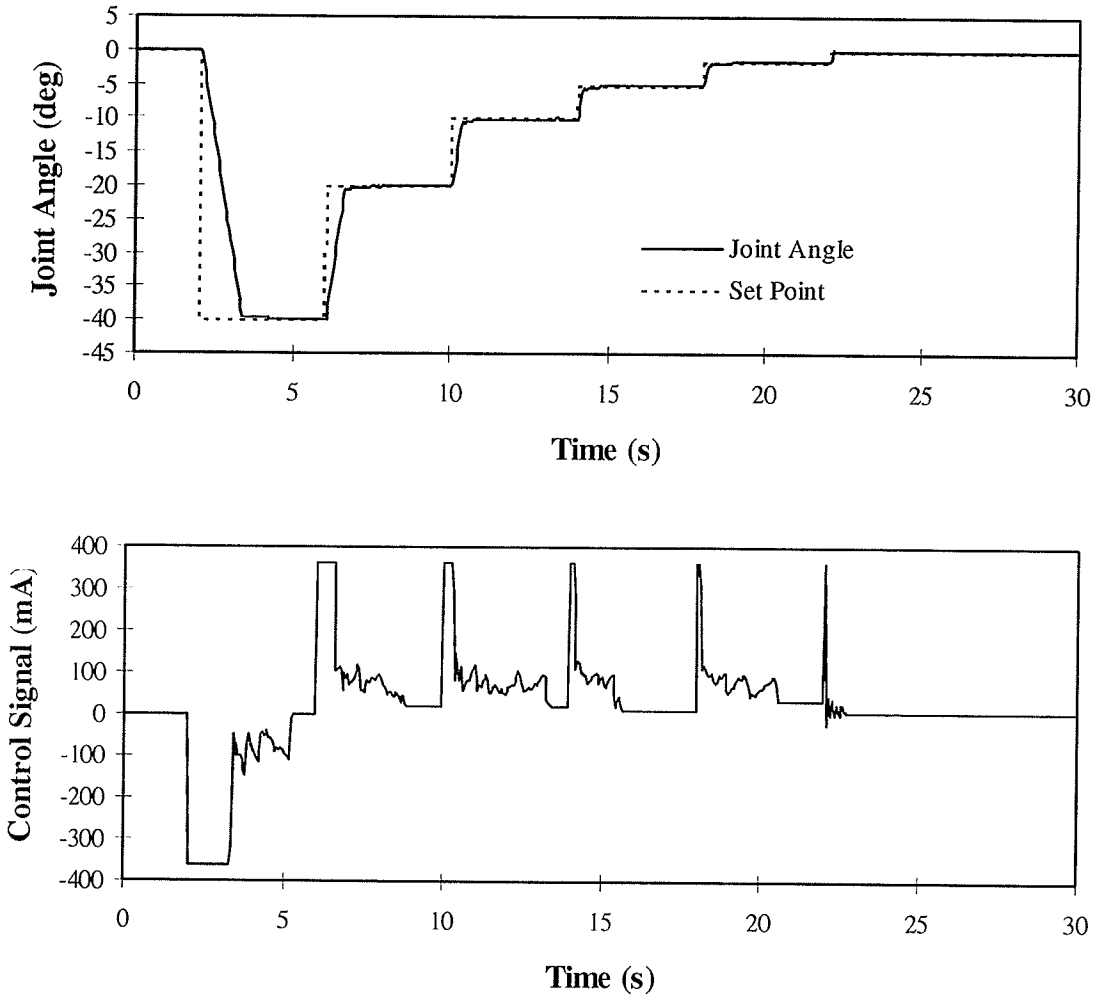


Figure 4-17: VETIA, large step pattern ($\dot{e}_{min}$ =0.15deg/s, U_{lower}=-88mA, U_{upper}=114mA).

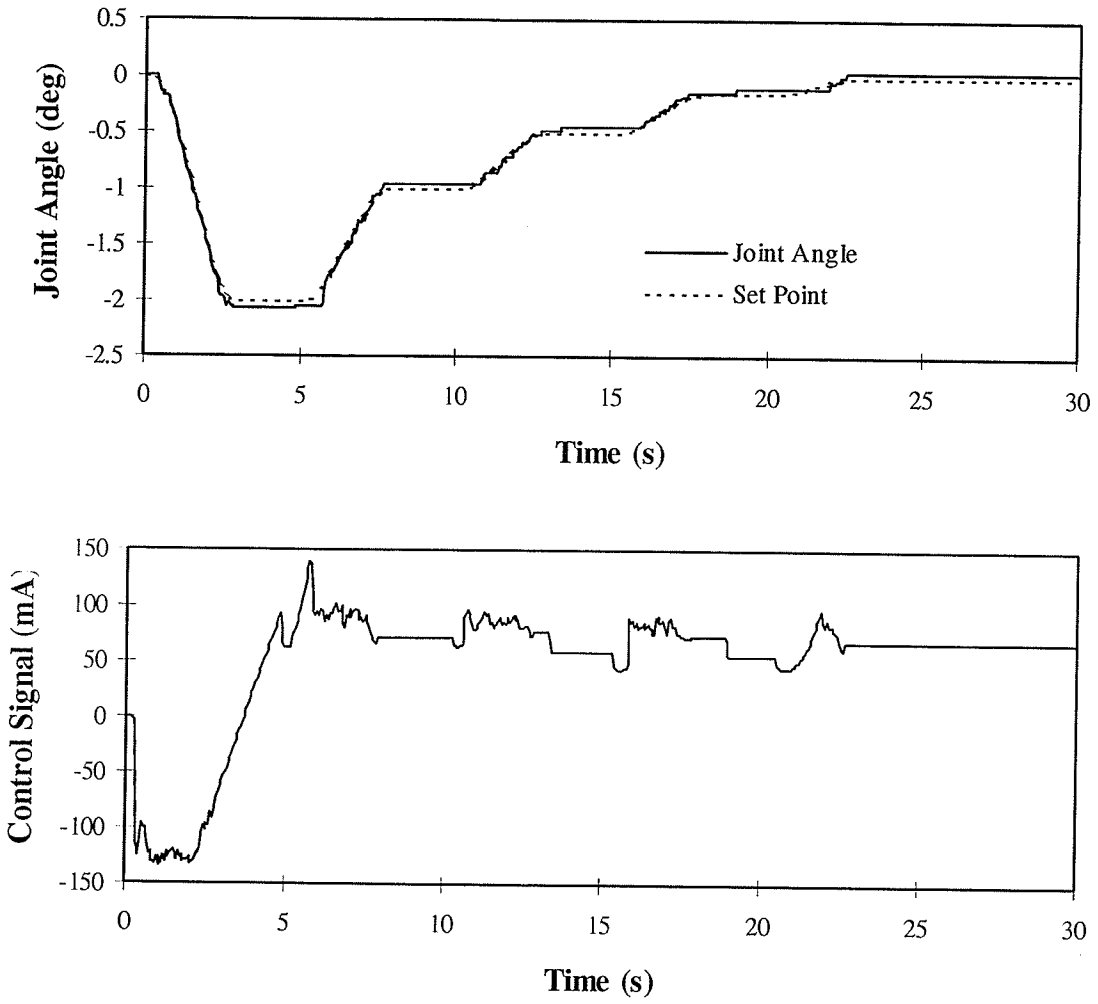


Figure 4-18: VETIA, small ramp pattern ($\dot{e}_{min}$ =0.15deg/s, U_{lower}=-88mA, U_{upper}=114mA).

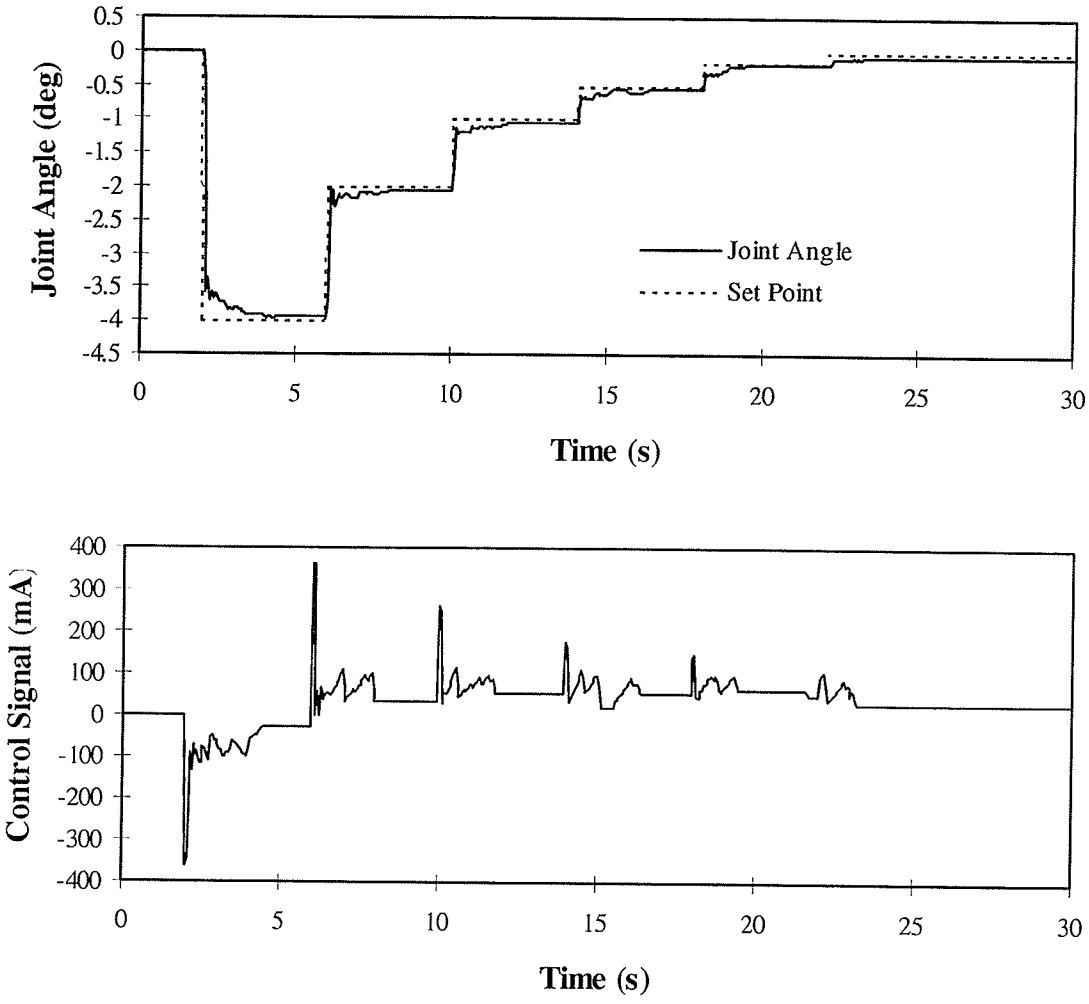


Figure 4-19: VETIA, small step pattern ($\dot{e}_{min}$ =0.15deg/s, U_{lower}=-88mA, U_{upper}=114mA).

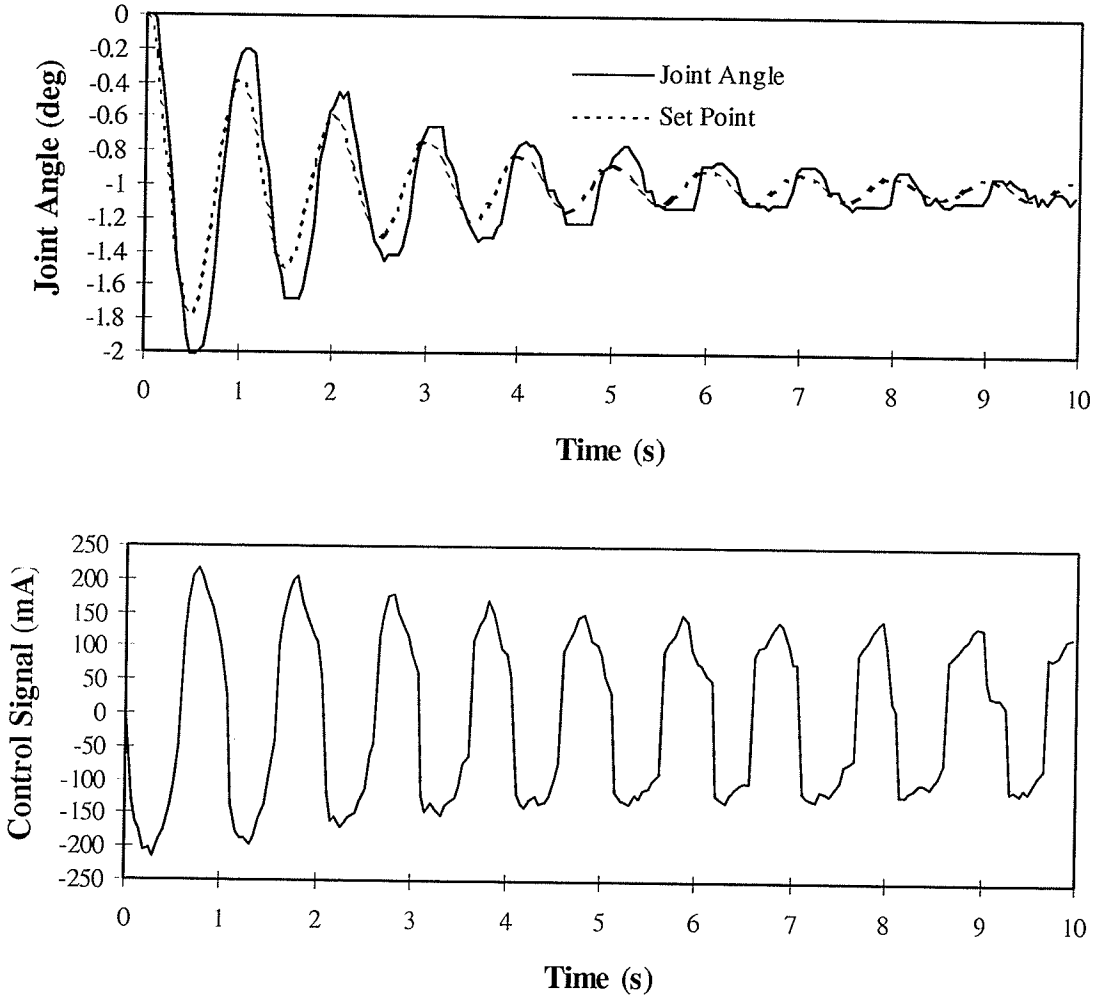


Figure 4-20: VETIA, decreasing sine wave test ($\dot{e}_{min}$ =0.15deg/s, U_{lower}=-88mA, U_{upper}=114mA).

4.6 Overshoot Reduction

Figure 4-20 shows how the proposed method of stiction compensation eliminates sticking. However, the same graph shows recurrent large overshoot in tracking a 1Hz sinusoidal input, particularly at larger amplitudes. The ramp response tests also show this tendency to overshoot. In force control applications, where accurate tracking of small motions must be achieved, such an overshoot is not permissible.

The cause of the problematical overshoot is the same as that of the excellent set-point tracking during the rest of the trajectory; namely, the integral portion of the controller. In order to maintain very small error, the control signal must be almost entirely due to the integral term. While allowing for excellent tracking at constant velocities, the integral is slow to change, resulting in poor tracking of velocity changes. Even with the addition of the rate-varying factor in the modified Morse controller, the integral is too slow to die down when the velocity set-point is brought to zero.

Some means is therefore needed to decrease the integral term when braking accelerations are specified. This was accomplished by including a set-point acceleration based term in the integral, i.e.

$$I(t) = \left(I(t-\Delta t) + e(t)\Delta t + K_a \ddot{q}^{braking}\Delta t\right)\frac{\alpha(\dot{q})}{\alpha(\dot{q}) + \dot{e}^2(t)}$$

$$\ddot{q}^{braking} = \begin{cases} \ddot{q} & \text{if } \ddot{q}\dot{q} < 0 \\ 0 & \text{if } \ddot{q}\dot{q} \geq 0 \end{cases}$$

where K_a is another controller gain. Without the nonlinear multiplicative factor $\frac{\alpha}{\alpha+\dot{e}^2}$, integrating the desired acceleration trajectory is completely equivalent to adding velocity feedback, provided that the desired and actual trajectories are practically identical, which has been demonstrated to be the case here. Limiting the accelerations integrated to braking accelerations will therefore reduce the integral when the desired velocity is approaching zero, which is when the problems with overshoot are observed. The

incorporation of the nonlinear factor can only assist in decreasing the magnitude of the integral.

No negative effects were observed as a result of this modification. The step responses were unaffected, and in no trial did the control signal reduction appear to reduce the speed of response.

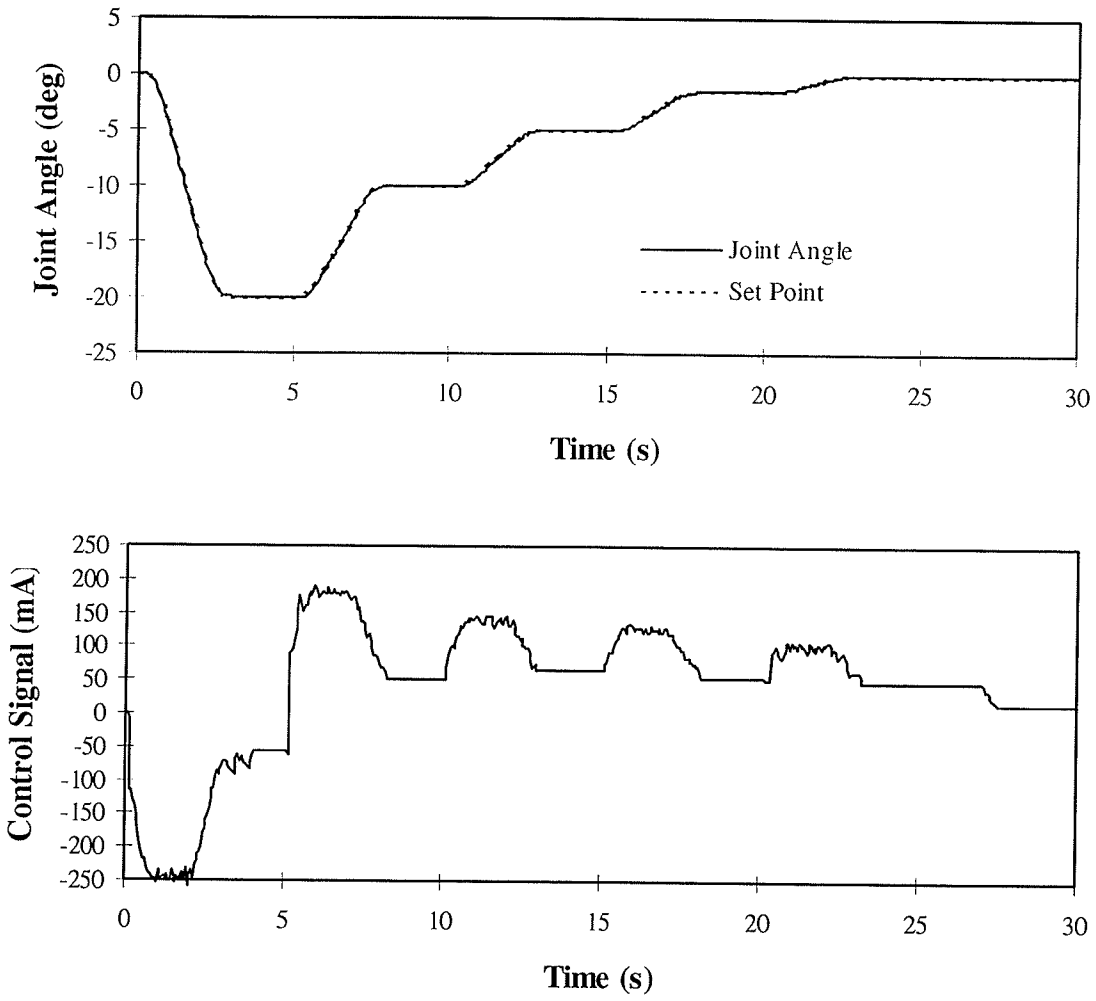


Figure 4-21: Final position controller, large ramp pattern (K_a=150s^2).

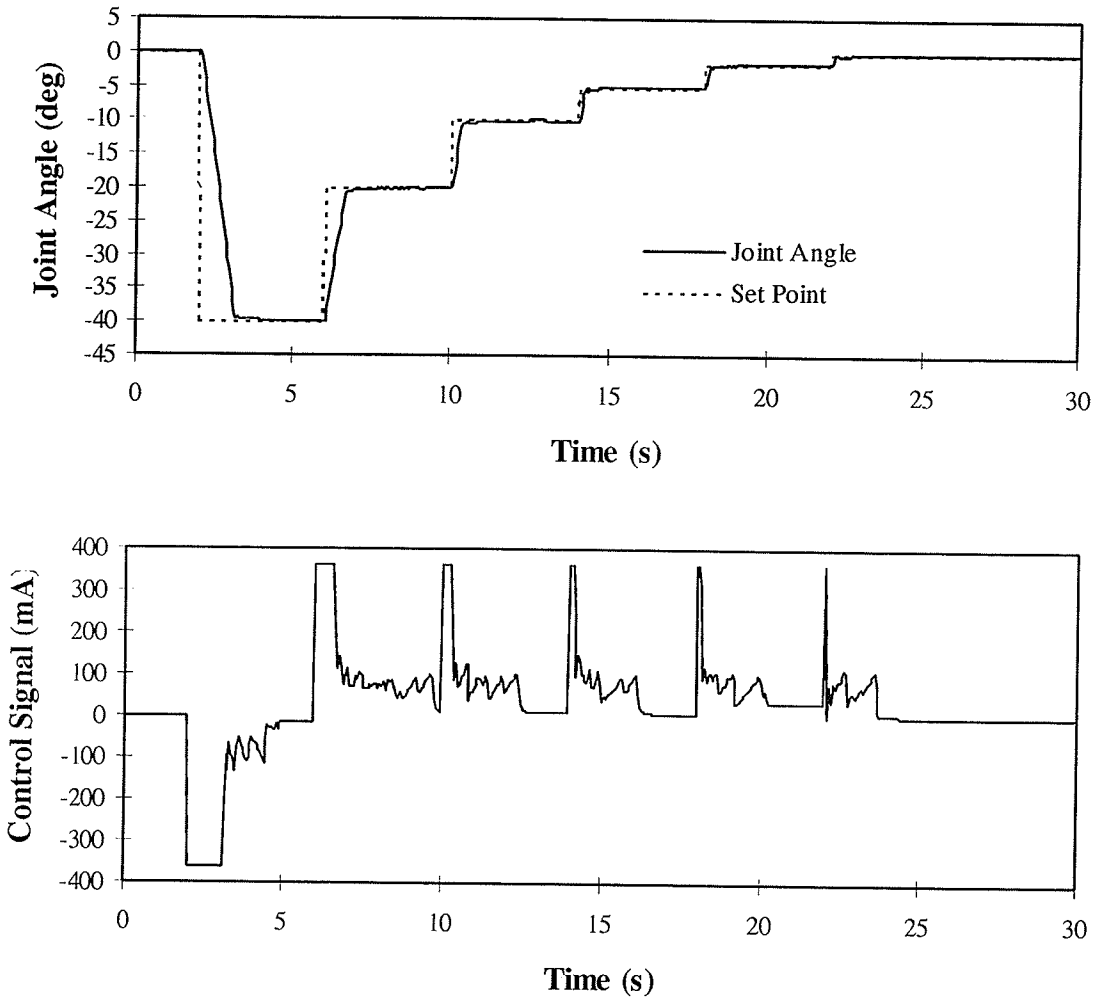


Figure 4-22: Final position controller, large step pattern (K_a=150s^2).

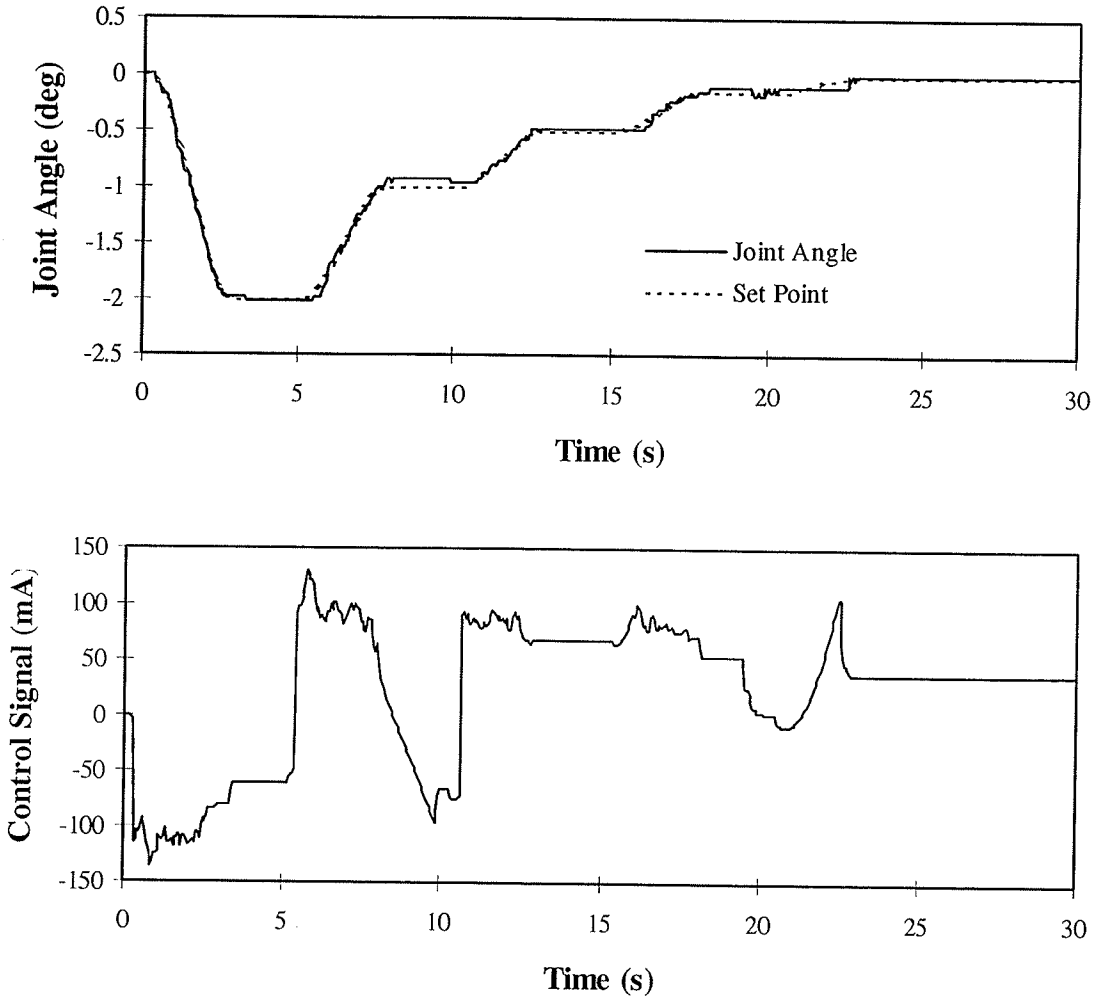


Figure 4-23: Final position controller, small ramp pattern (K_a=150s^2).

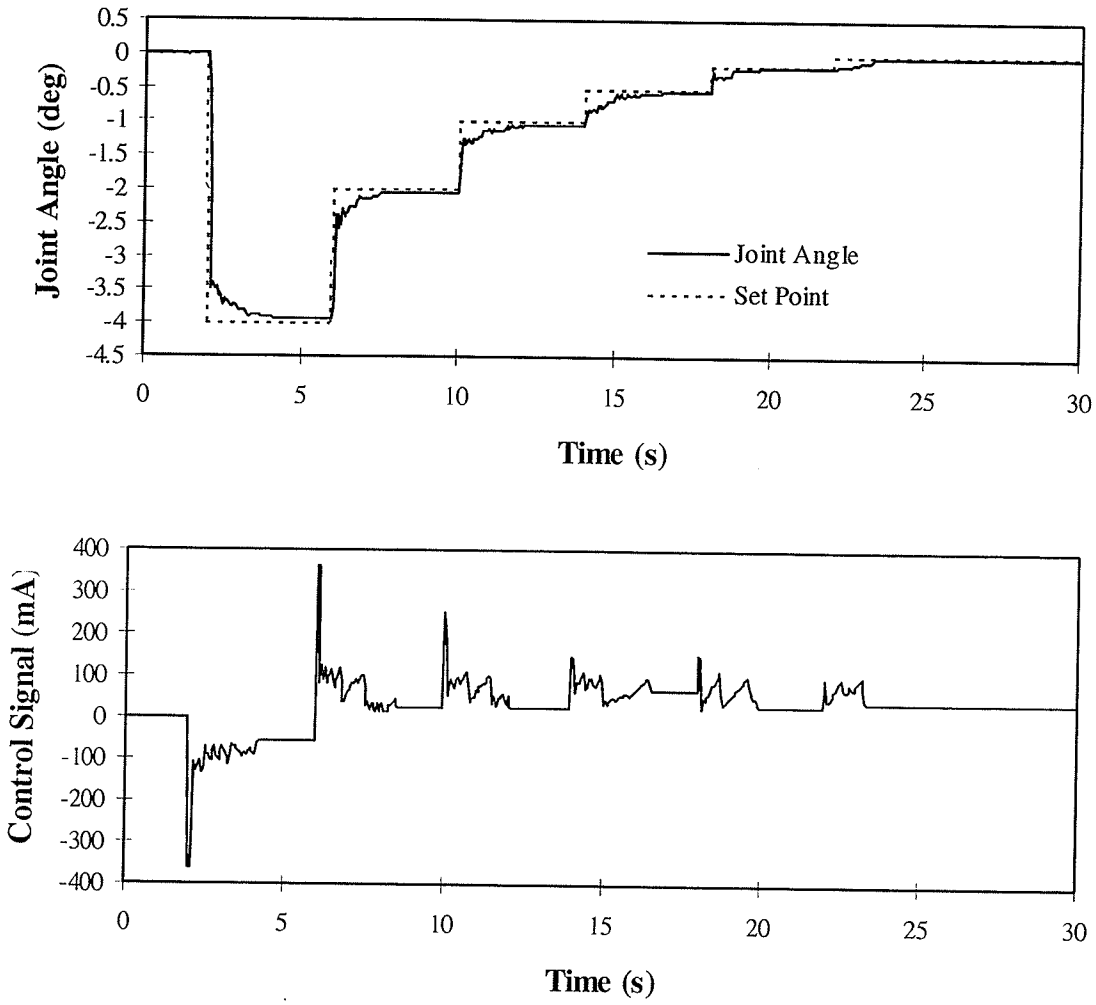


Figure 4-24: Final position controller, small step pattern (K_a=150s^2).

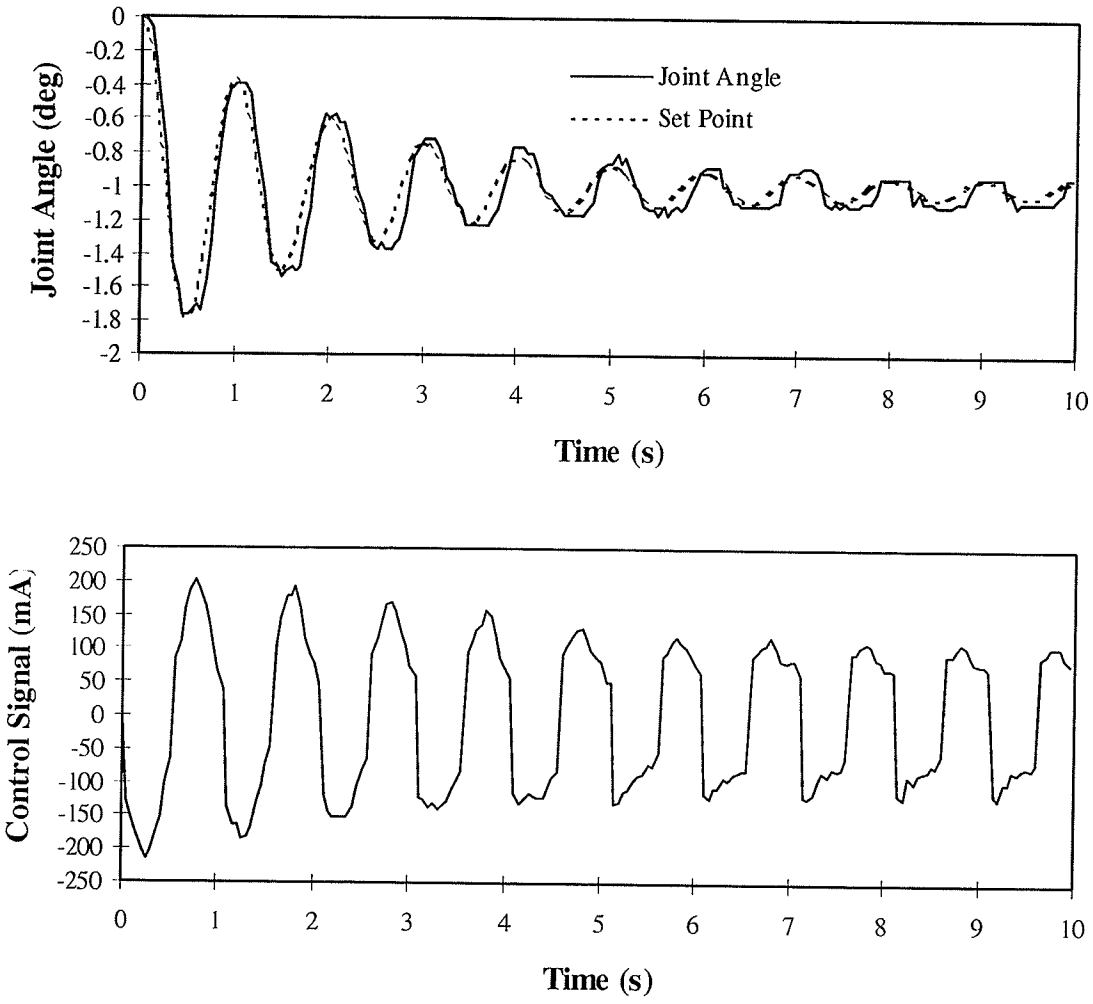


Figure 4-25: Final position controller, decreasing sine wave test (K_a=150s^2).

Figure 4-25 gives the most striking illustration of the improvement in set-point tracking resulting from the final modification. The manipulator could now follow the sine wave closely for the larger amplitude oscillations. The control signal had a scalloped appearance, being diminished by the additional term whenever a deceleration was specified. The same effect provided some improvement in tracking ramp inputs as well.

The remaining area for improvement is in tracking very low amplitude changes in the set-point. Figure 4-25 shows that the controller still has difficulty following the sine wave once the peak-to-peak amplitude drops to 0.2°. This amplitude corresponds to

about 7 times the encoder resolution. Tracking such small changes accurately while retaining stability in the face of large changes in set-point is a very difficult task because the effects of stiction and actuator deadband are most apparent in this region of operation. The ramp and step trials show that the steady-state positioning error is about 1 or 2 encoder resolution widths. These errors were concluded to be the limits on the positioning ability of this controller.

4.7 Summary

Three nonlinear modifications to a simple linear PI control law, as applied to an industrial hydraulic manipulator, increased its high-frequency low-amplitude tracking abilities by an order of magnitude. The resultant controller was able to accurately track position trajectories down to 7 encoder resolutions of error, and regulate static trajectories to within 1 encoder resolution. These abilities were displayed in spite of large manipulator inertia, joint stiction and actuator deadband.

5. Impedance Control Tests

5.1 Experimental Setup & Strategy

5.1.1 Test Plate

Three separate environments were used separately or in tandem to test the impedance controller: free space (infinite admittance), an aluminum plate (variable linear admittance), and a steel box resting on a wooden table (nonlinear admittance). Most experiments used the thin, simply-supported aluminum plate. By varying the support spacing, the environmental stiffness could be varied over an order of magnitude (see Appendix B). Tests involving the plate were repeated using stiffnesses of 7.2kN/m and 53kN/m. The box on the table was used for tests of impact with an inertial body subject to Coulomb friction ($\mu_{static} \approx 0.37$, $\mu_{kinematic} \approx 0.29$). It was filled with 64kg of weights.

5.1.2 Sensor

The force sensor was based on a standard strain gauge bridge. It had a range of measurement in a single direction of 0 to 700N with ±10N of noise. This level of noise is due to transmission from the sensor to the computer, and is not atypical (Field and Stepanenko, 1993). No attempt was made to reduce the level of noise in the reading because of the intended industrial application of the controller, where such idealizations may not be feasible. Any force bias in the sensor readings, however, was removed in software by 2 seconds of sampling before each test.

The force sensor was mounted in the configuration shown in Figure 5-1. The primary disadvantage of the mounting was that the gripper was not able to hold the assembly rigidly about its own wrist axis. As a result, the force sensor was subject to some considerable latitude in the X-direction when positioned to measure vertical forces; gripper rotations of up to 5° were observed in the following experiments. This latitude accounts for some of the scatter in measurement of the sensor stiffness, which was determined to be 46kN/m (see Appendix E).

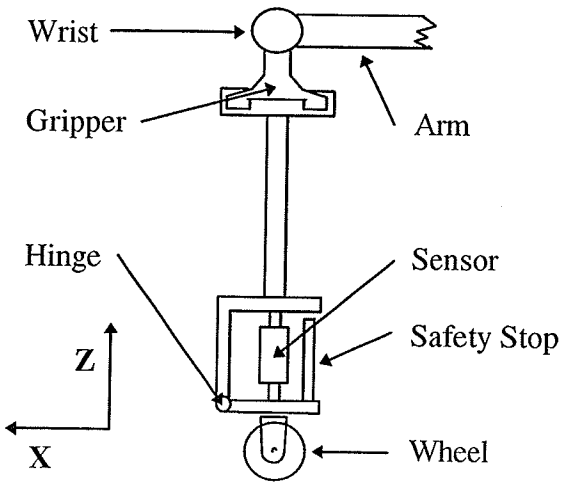


Figure 5-1: Force sensor mounting assembly.

5.1.3 Controller Implementation

The impedance controller was built as an outer loop about the position controlled system developed in Section 2.1. A diagram of the controller has been shown in Figure 3-4. The impedance control loop accomplished the following steps:

1) the nominal Cartesian position trajectory was calculated,
2) this trajectory was modified as required by the target impedance, using force feedback,
3) the inverse kinematics were used to transform the modified position trajectory into joint space, and
4) the joint-space trajectory was passed to the position controller in the inner loop.

For a second order linear target impedance, the governing differential equation is:

$$m\ddot{x}_s + c(\dot{x}_s - \dot{x}_d) + k(x_s - x_d) = f_d - f$$

The meaning of each quantity in the above equation is apparent in Figure 3-6. The trajectory modification was calculated at each iteration according to the known analytical

solution of the models response to a step input. For example, if the model represented an underdamped system, the force response was calculated as follows:

$$x_s(t_i) = e^{-\xi\omega_n \Delta t_i}\left\{A_i \cos(\omega_d \Delta t_i) + \frac{B_i}{\omega_d}\sin(\omega_d \Delta t_i)\right\} + \frac{(f_d)_i - f_i}{k}$$

$$A_i = x_s(t_{i-1}) - \frac{(f_d)_i - f_i}{k},\ B_i = \dot{x}_s(t_{i-1}) + \xi\omega_n A_i,\ \Delta t_i = t_i - t_{i-1}$$

where ω_n is the natural frequency for the specified target mass, damping, and stiffness, ξ is the corresponding damping coefficient, and $\omega_d=\omega_n(1-\xi^2)^{0.5}$. The sampling time Δt_i averaged 4.5ms, justifying the assumption of quasi-static forces.

The poles of the position response transfer function are the same as those of the force response, but there is a zero at s = -k/c. Therefore, the response to a step change in x_d would be calculated using the same equations as above, except that the constants A_i and B_i are calculated as follows:

$$A_i = x_s(t_{i-1}) - x_d(t_i),\ B_i = \dot{x}_s(t_{i-1}) + \xi\omega_n\left[A_i - 2\left(x_d(t_i) - x_d(t_{i-1})\right)\right]$$

5.1.4 Overview of the Experiments

In the following tests of the position based implementation of impedance control, certain general considerations influenced the selection of the target impedance parameters. A large mass was typically selected (50 to 500kg) in order to have system dynamics that were reasonable for the position controller to track (no frequencies greater than 1Hz). Large masses also served to filter the noise in the force sensor readings. Values for the target stiffness were selected in order to obtain both the desired response time and the desired compliance of the system. When damping ratios are referred to, they are either the damping ratio ξ or the net damping ratio ξ_{net} of the composite model-sensor-environment system, assuming pure stiffness in sensor and environment k_{eff}. These damping ratios are calculated as:

$$\xi = \frac{c}{2\sqrt{mk}},\ \xi_{net} = \frac{c}{2\sqrt{m\left(k + k_{eff}\right)}}$$

$$\xi = \xi_{net}\sqrt{1 + \frac{k_{eff}}{k}}$$

Five different sets of experiments were performed in order to assess the performance of the impedance controller. These experiments tested:

1) free space force response,

2) static force control,

3) dynamic force control,

4) interrupted position trajectory tracking, and

5) pushing a heavy object against stick-slip friction.

For each of the different experiments, three quantities were plotted: force, position, and control signal. Force set-points refer to the reference signal F_d. Position set-points refer to the modified position trajectory x_s used as input to the position controller. When it appears, the nominal position trajectory is the unmodified trajectory x_d input to the impedance controller. The reader is referred to Figure 3-4 & Figure 3-6 for an illustration of the relationship between these variables.

A convention must be explained in order to understand the control signal plots. The control signal plotted is the one sent to the up/down link of the Unimate for all experiments except the inertial obstacle tests, where the control signal for link 3 (in/out motion) is plotted. These signals best reflected the control action taken along the degree-of-freedom in which forces are encountered. It is important to note that the up/down control signal was positive when producing a downward motion because of conventions in joint description (see Appendix A). The in/out link had no such reversal, so its control action was taken in the same sense as the corresponding Cartesian motions.

5.2 Free Space Response

The free space response involved a step change in the force set-point when the end-effector was far from any obstacles. The purpose of this set of tests was to ensure that the apparent manipulator impedance matched the target impedance under a simple step input. Damping was varied to obtain overdamped, underdamped, and critically damped responses.

The response of the impedance-controlled system to a step change in the forces applied to the end-effector is shown for various impedance parameters in Figure 5-2 through to Figure 5-4. In each case, the position controller successfully coordinates the motions of each link to make the end-effector follow the required Cartesian trajectory with great precision.

The impedance controller successfully replaces the existing manipulator dynamics with the specified target impedance. The steady-state change in position observed in all trials is the correct 0.2m for a 100N step in force with a target stiffness of 500N/m. Furthermore, Figure 5-2 shows that the period of oscillation of the underdamped system matches the desired 3.3 seconds.

The only problem observed in this experiment was difficulty in tracking the dying oscillations of the underdamped system. This problem is not serious, since the desired motions are very small (< 1mm), and underdamped responses are unlikely to be specified in practice.

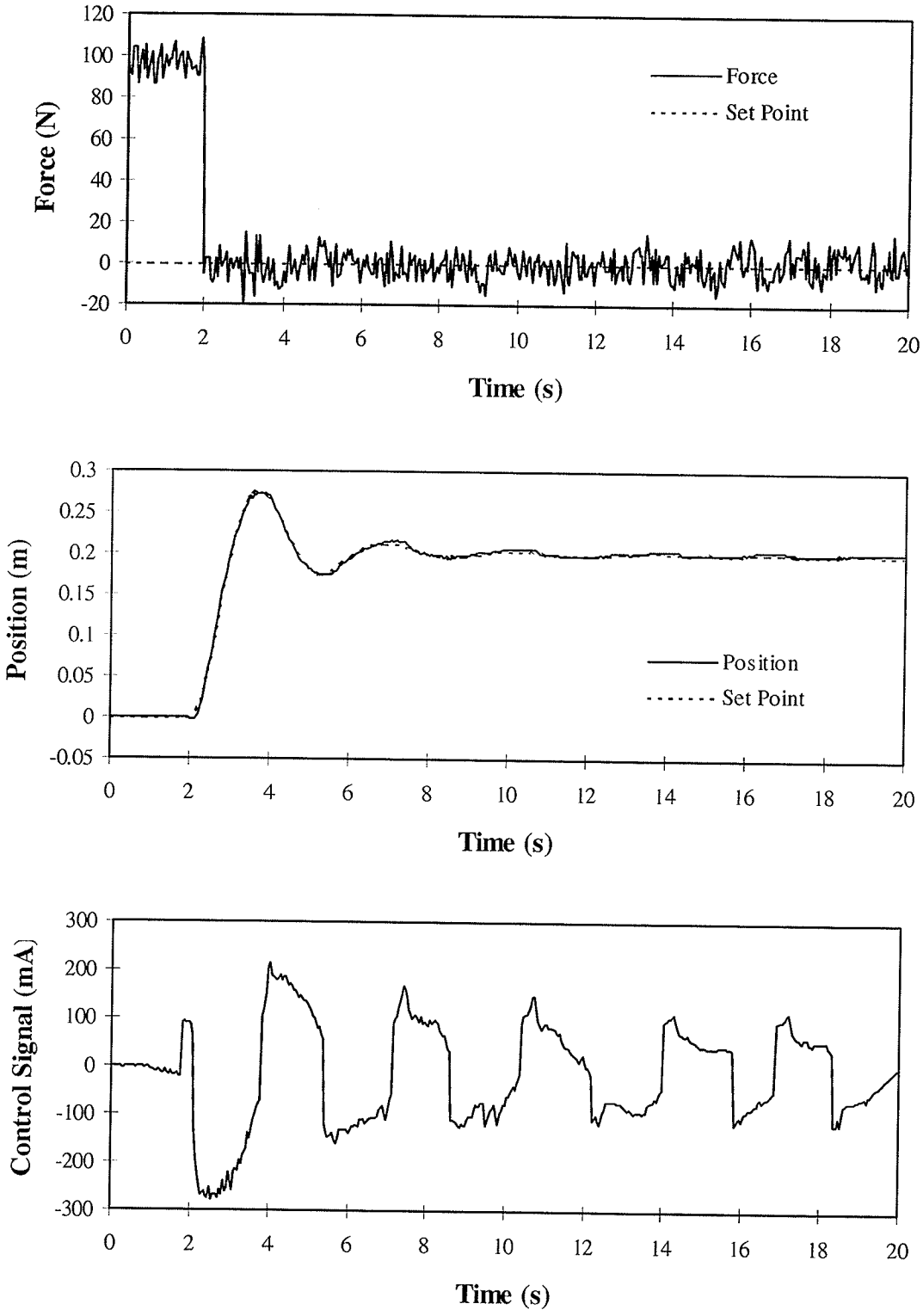
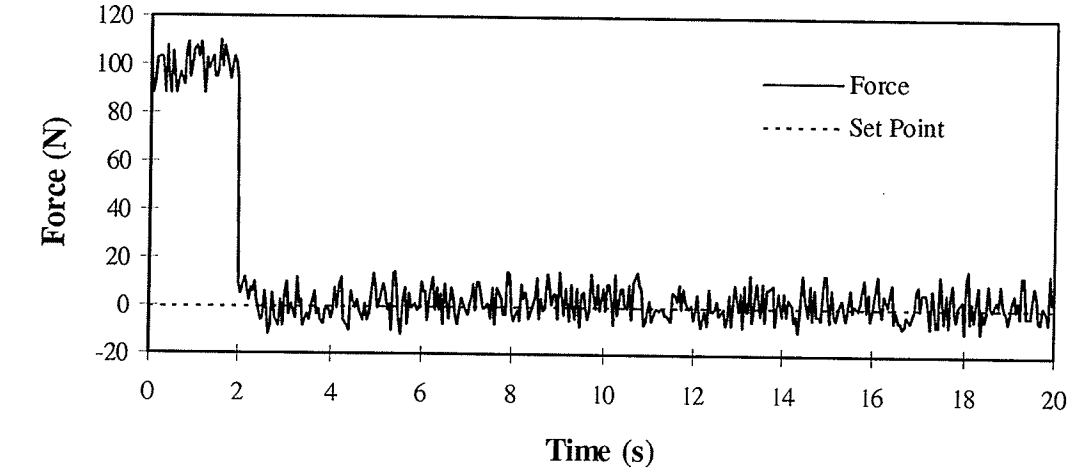
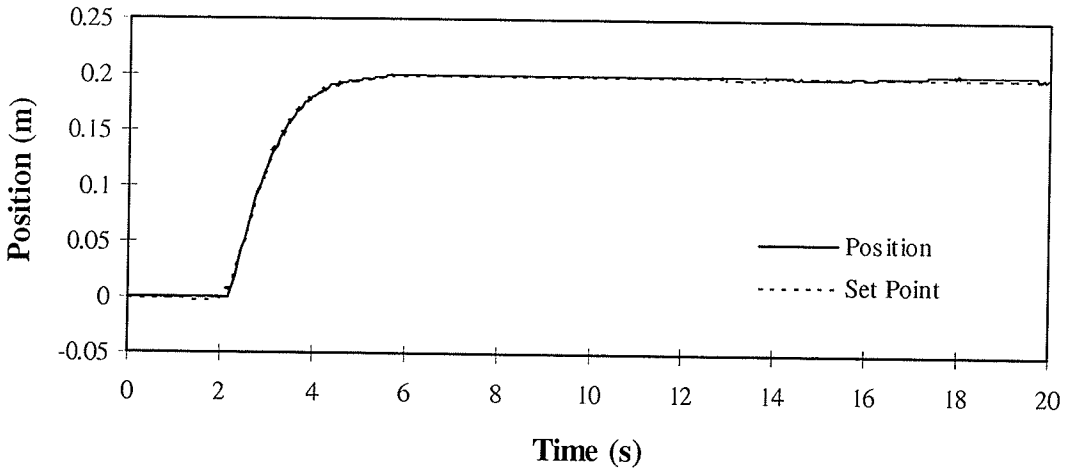


Figure 5-2: Free space response, m=125kg, ξ=0.3, k=500N/m.





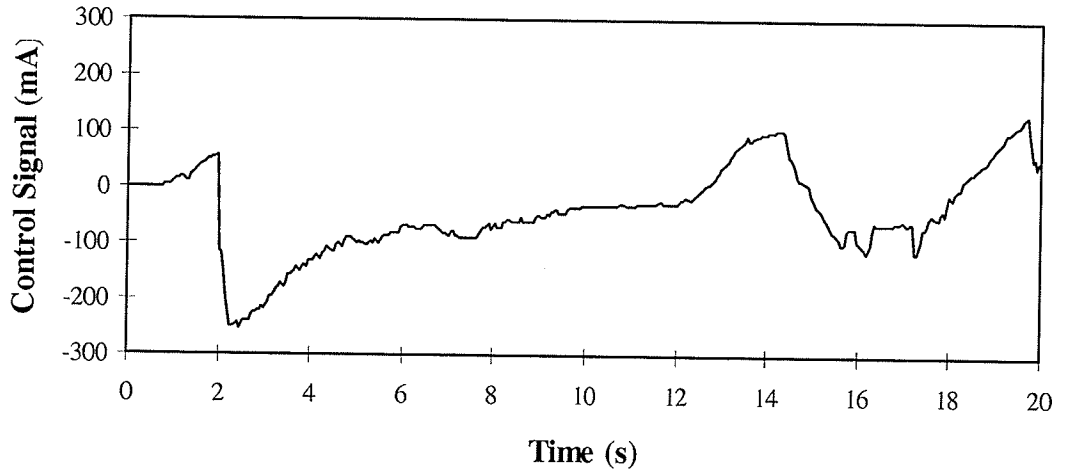


Figure 5-3: Free space response, m=125kg, ξ=1.0, k=500N/m.

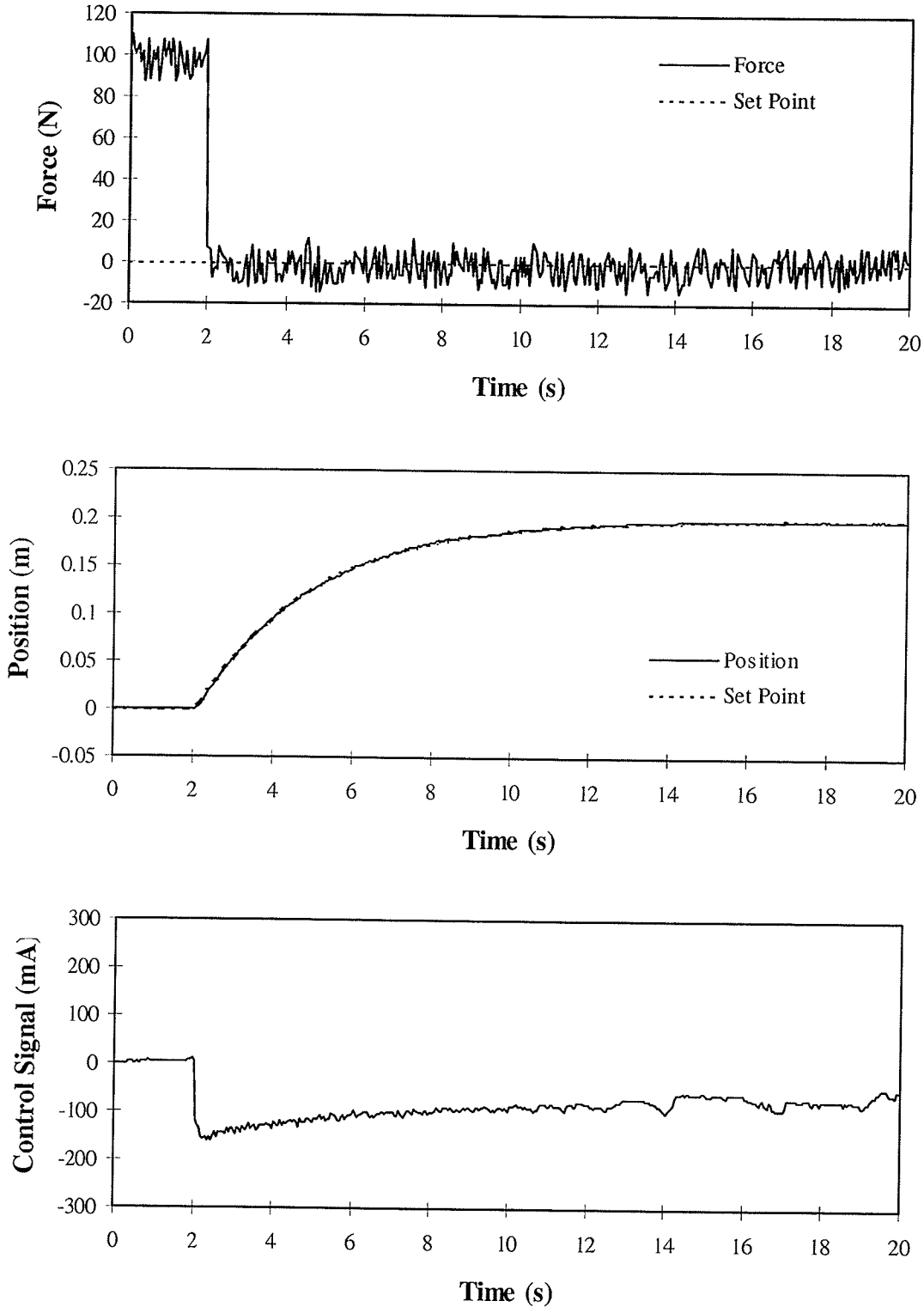


Figure 5-4: Free space response, m=125kg, ξ=3.0, k=500N/m.

5.3 Static Force Control

The static and dynamic force control tests were conducted using the test plate. The end-effector was positioned to be firmly in contact with the plate, then the force set-point was changed in either a step or 0.1Hz sine wave, as appropriate. The purpose of these tests was to determine the force regulation and tracking abilities of the impedance controller. To test the sensitivity of the controller to environmental stiffness, each test was repeated for k_e = 7.2kN/m and 57.6kN/m. Note that when combined with the sensor stiffness of 46kN/m, these values of k_e correspond to effective environmental stiffnesses k_{eff} of 6.2kN/m and 25kN/m, respectively. These force control tests and conditions are similar to those used by others (see Volpe and Khosla, 1993b; Dawson, Lewis and Dorsey, 1992; Liu and Goldenberg, 1991).

As discussed in Section 6.3, for best steady-state accuracy in static force control, the target stiffness must be as low as possible. However, is will be shown in Section 7 that low target stiffnesses result in eventual limit cycle behaviour. Therefore, high target stiffnesses and some form of compensation must be used. Therfore, the impedance parameters selected were m=500kg, ξ_{rel}=1.2, and k=50kN/m, and were held constant across all force control tests. These parameters were selected to obtain a fast response and long-term stability (see Section 7). However, the large target stiffness required the use of a set-point compensation scheme to counteract the errors stemming from load sharing between the model and the physical system. The force set-point was simply increased by a gain of $kk_{eff}/(k+k_{eff})$. This gain is the exact amount required to offset load-sharing and regain the correct steady-state force (see Section 6.2).

This experiment shows that this force set-point compensation scheme can be used for increasing accuracy in static force control. Comparing the steady-state accuracy displayed in Figure 6-10 and Figure 5-5 illustrates this point: Figure 5-5 shows an error of approximately 6% with a target stiffness of 50kN/m when using set-point compensation, which is comparable to the results obtained using a stiffness of 500N/m without compensation. The remaining error shows the requirement for accuracy in the estimate of

environmental stiffness required by the technique, however. Figure 5-6 shows an error closer to 10%, showing that the technique can still be used in stiff environments, but remains sensitive to the estimated stiffness. The alternative to this error is the limit cycle behaviour resulting from using low target stiffnesses (see Section 7). The control signal in both environments showed no sign of inducing any further motions.

On the whole, this experiment shows that position based impedance control can be used for static force control in both stiff and compliant environments.

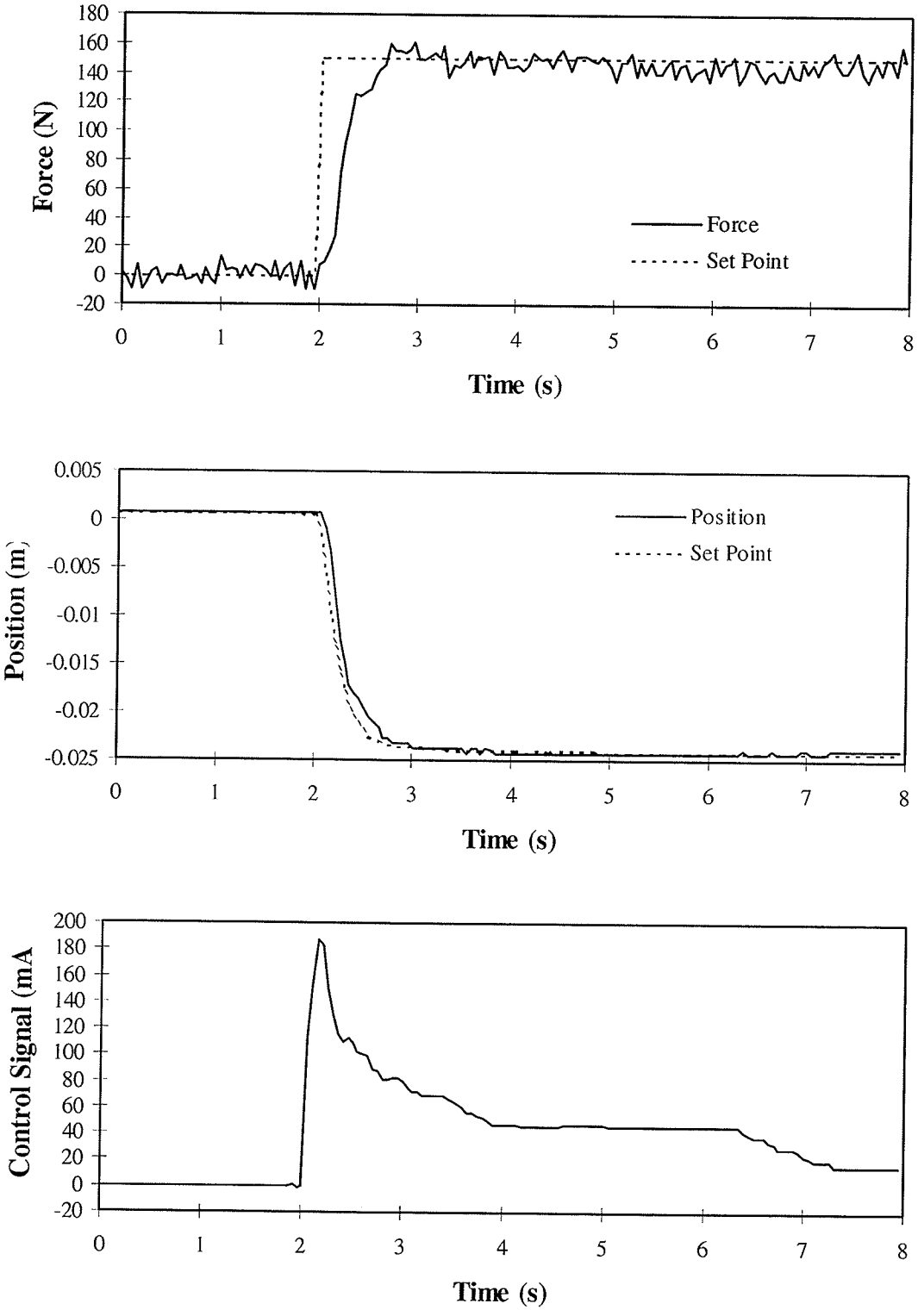


Figure 5-5: **Static force control with compensation, low environmental stiffness (m=500kg, ξ_{net}=1.2, and k=50kN/m).**

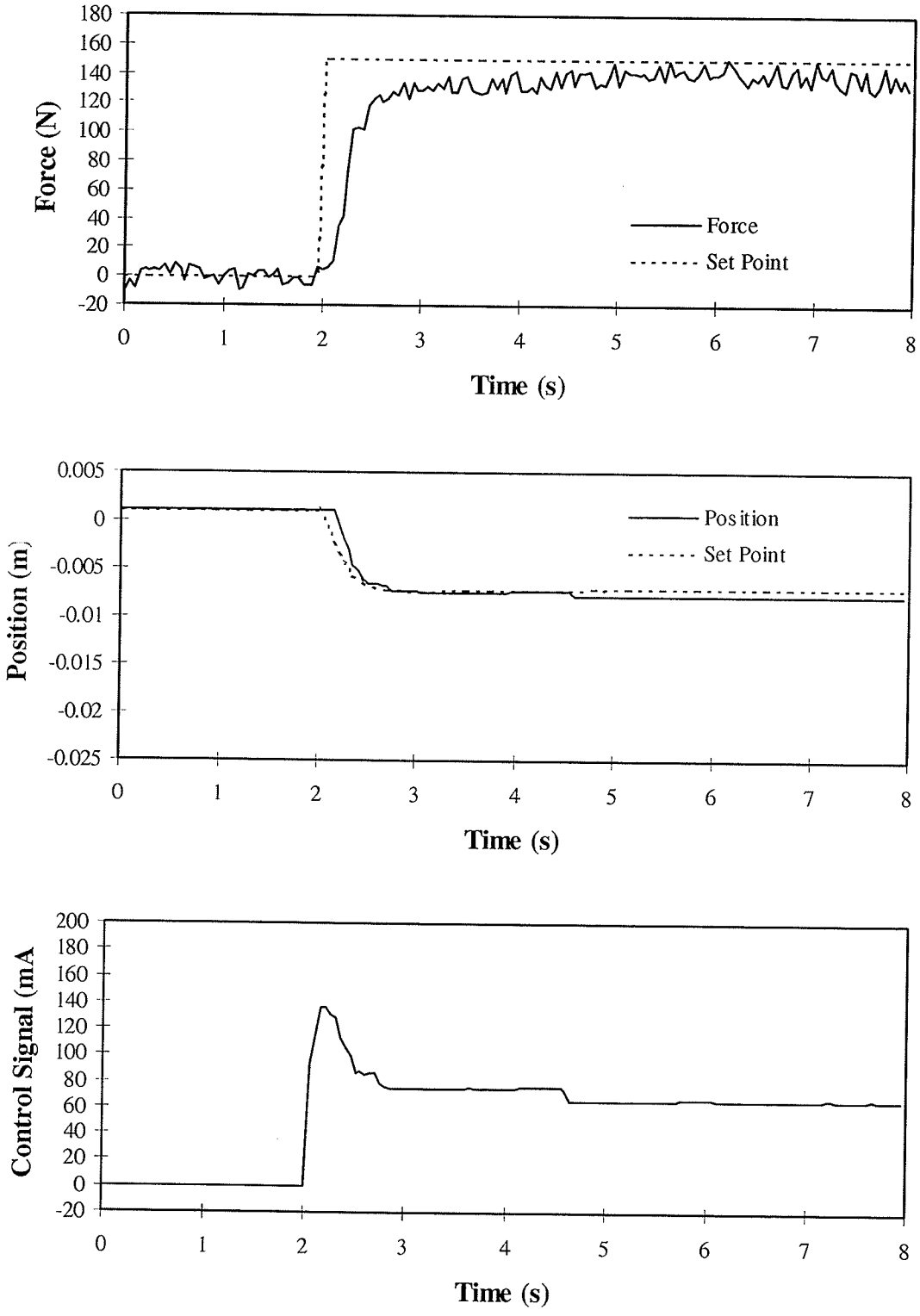


Figure 5-6: **Static force control with compensation, high environmental stiffness (m=500kg, ξ_{net}=1.2, and k=50kN/m).**

5.4 Dynamic Force Control

This experiment shows that position based impedance control with set-point compensation can be successfully applied to a dynamic force control task, but performs best in a compliant environment. Figure 5-7 illustrates this performance, which is obtained in spite of the fact that the force set-point compensation method only compensated for the static error in the effective force set-point (see Section 6). The target impedance dynamics are fast enough that no transient response interferes with force tracking. The target impedance parameters were the same as those used for the static force control experiment, showing some versatility in application without adjusting the control parameters.

Tracking was poorer in the stiffer environment, as shown in Figure 5-8. The primary difficulty in tracking the force set-point was in tracking the very small required position set-point changes. The motions required in the stiff environment were of less than 1_{cm} peak-to-peak, corresponding to less than 0.6° total change in joint angle on the up/down link. The position controller proved able to track even these small changes while moving down, but upward motions were less controlled. This is because the end-effector forces were sufficiently high to move the manipulator without assistance from the actuators. The manipulator would still not move without any control action because the hydraulic actuators were very stiff when the spool valve was shut. However, with the valves open, the dynamics required that the actuators apply a small torque against the desired direction of motion. The controller, however, was designed on the supposition that the control action required would always be in the same direction as the position error. This is the cause of the highly oscillatory behaviour of the control signal observed during upwards motions under large forces in both environments (best seen between 7 and 9 seconds in Figure 5-8).

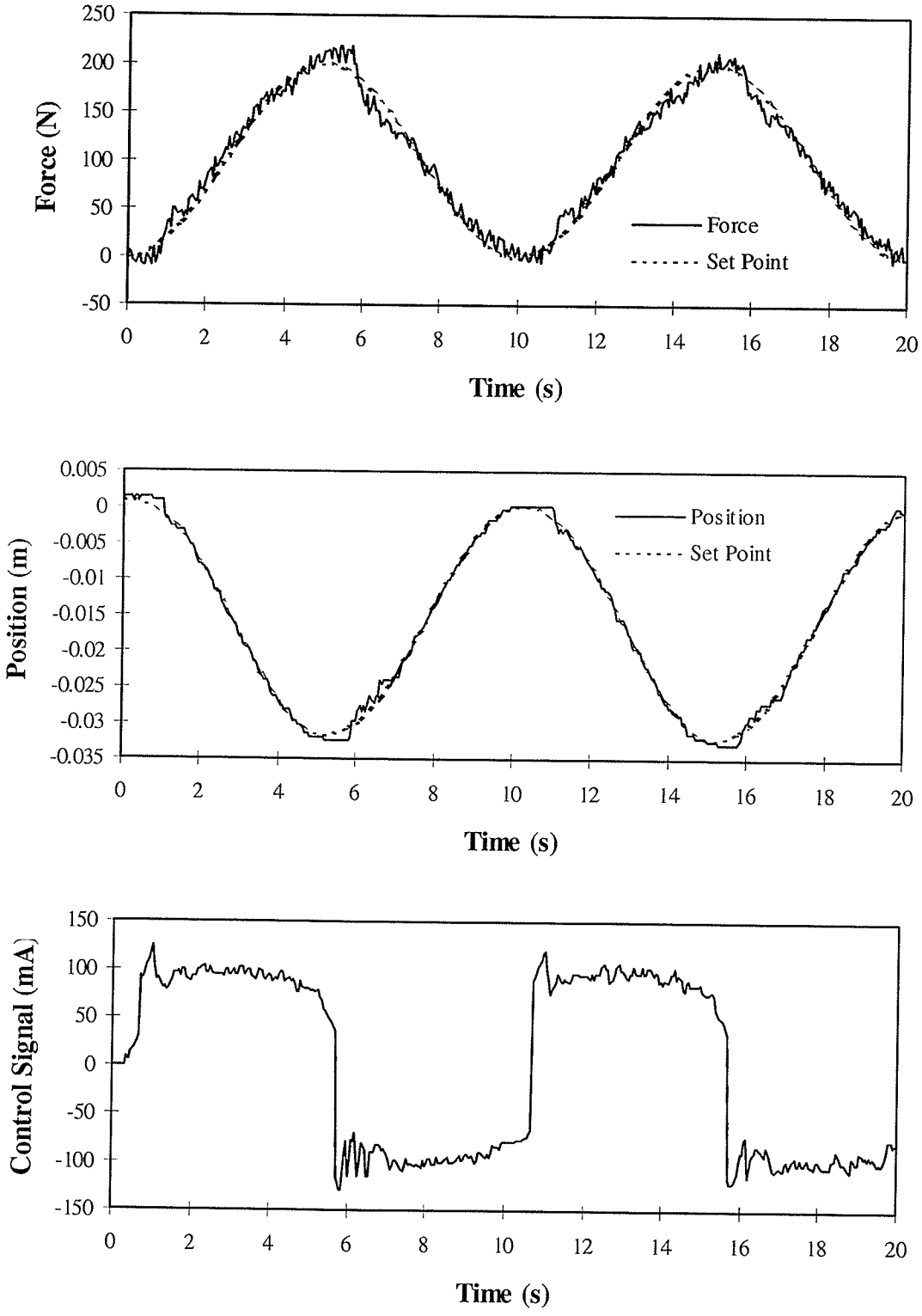


Figure 5-7: **Dynamic force control with compensation, low environmental stiffness (m=500kg, ξ_{net}=1.2, and k=50kN/m).**

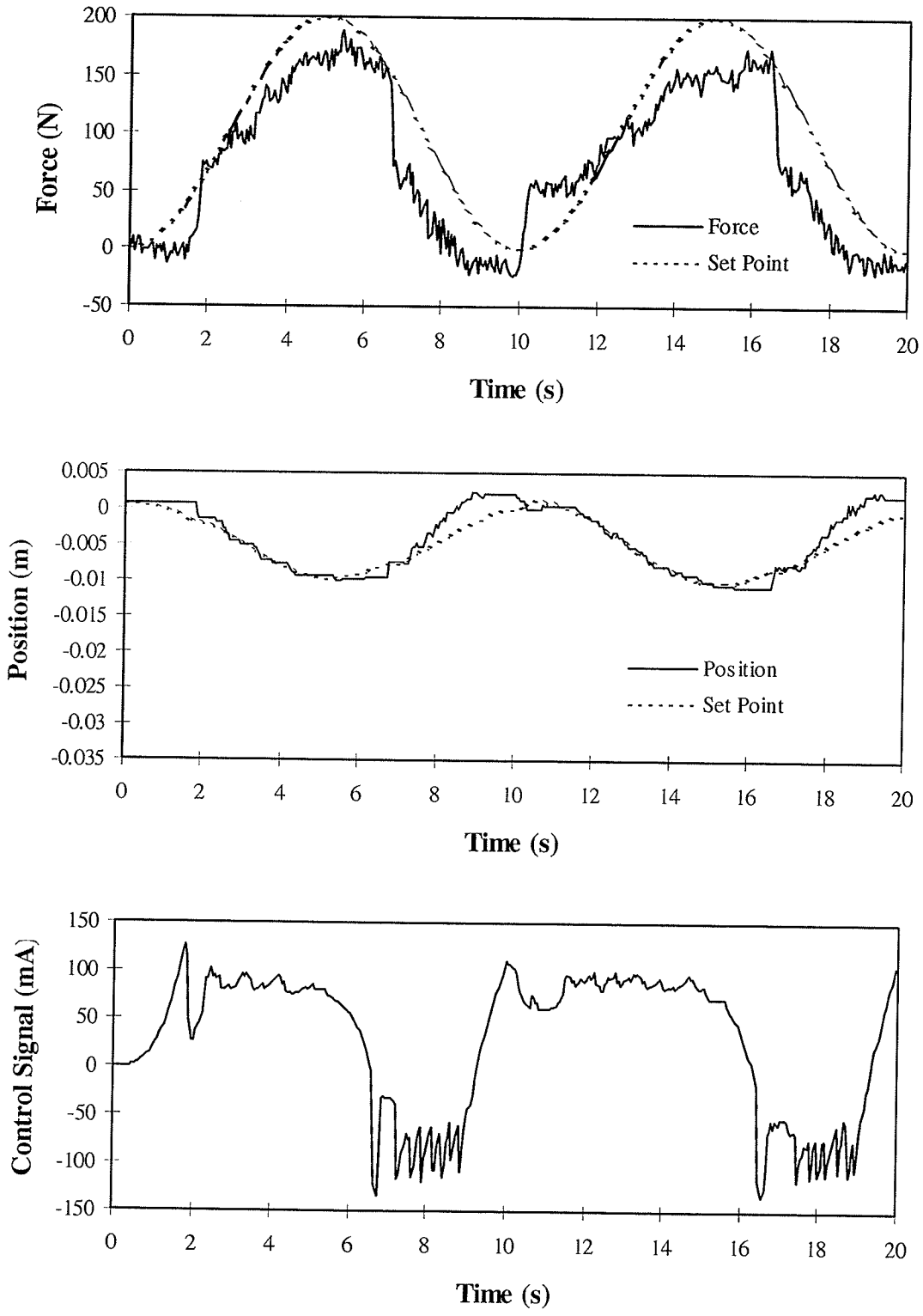


Figure 5-8: **Dynamic force control with compensation, high environmental stiffness (m=500kg, ξ_{net}=1.2, and k=50kN/m).**

5.5 Springy Obstruction of 2-D Position Trajectory

The fourth set of tests entailed commanding the end-effector to follow a circular trajectory in the X-Z plane. However, the plate was placed at Z=0, cutting the nominal trajectory circle in half. This provided a test of the controller's response to sudden and large changes in environmental admittance, as well as testing its ability to control position in one direction and deal with forces of interaction in another. This test has also been widely used (Hogan (1987); Lu and Meng (1991); Tsia and Orin(1986)). Again, the test was repeated using the same two configurations of the test plate, providing almost an order of magnitude difference in environmental stiffness.

Impedance parameters had to be set differently for the interrupted trajectory test than for the previous two because the task requirements were different. Instead of controlling forces, the goal was to minimize the forces of interaction while remaining as close to the nominal position trajectory as possible. This required a low target stiffness and fast dynamics. The settings used in both environments were m=50kg, ξ_{nel}=1.0, and k=1kN/m.

This experiment shows how the impedance controller deals with unexpected collision with a spring-like surface. The nominal position trajectory was quickly matched in free space, but encountering the surface at Z = 0 required trajectory modification. Figure 5-9 through Figure 5-12 show that the impedance controller effectively limited the forces of interaction while minimizing departures from the nominal trajectory. Limiting forces required a low target stiffness, while minimizing departures from the nominal trajectory required that transient responses die quickly, i.e. fast target dynamics. Given that a certain minimum target mass is required to filter the noise in the force readings, these are conflicting requirements. The settings used here represent a good compromise; forces were below 160N even in the stiff environment, and the transient response to environmental forces was of less than 5cm in magnitude, and died completely before the plate was struck again.

The controller demonstrated no instability upon striking the plate, even at its stiffest setting. This favourable behaviour was due to both the choice of impedance parameters and the compliance in the sensor. The target impedance selected did not over-react to the sudden contact forces, and the compliance in the sensor acted to reduce the magnitude of those forces. Some oscillation in control signal and force were observed just before losing contact with the plate (Figure 5-10 & Figure 5-12). This behaviour is due to the relative values of target and environmental stiffness.

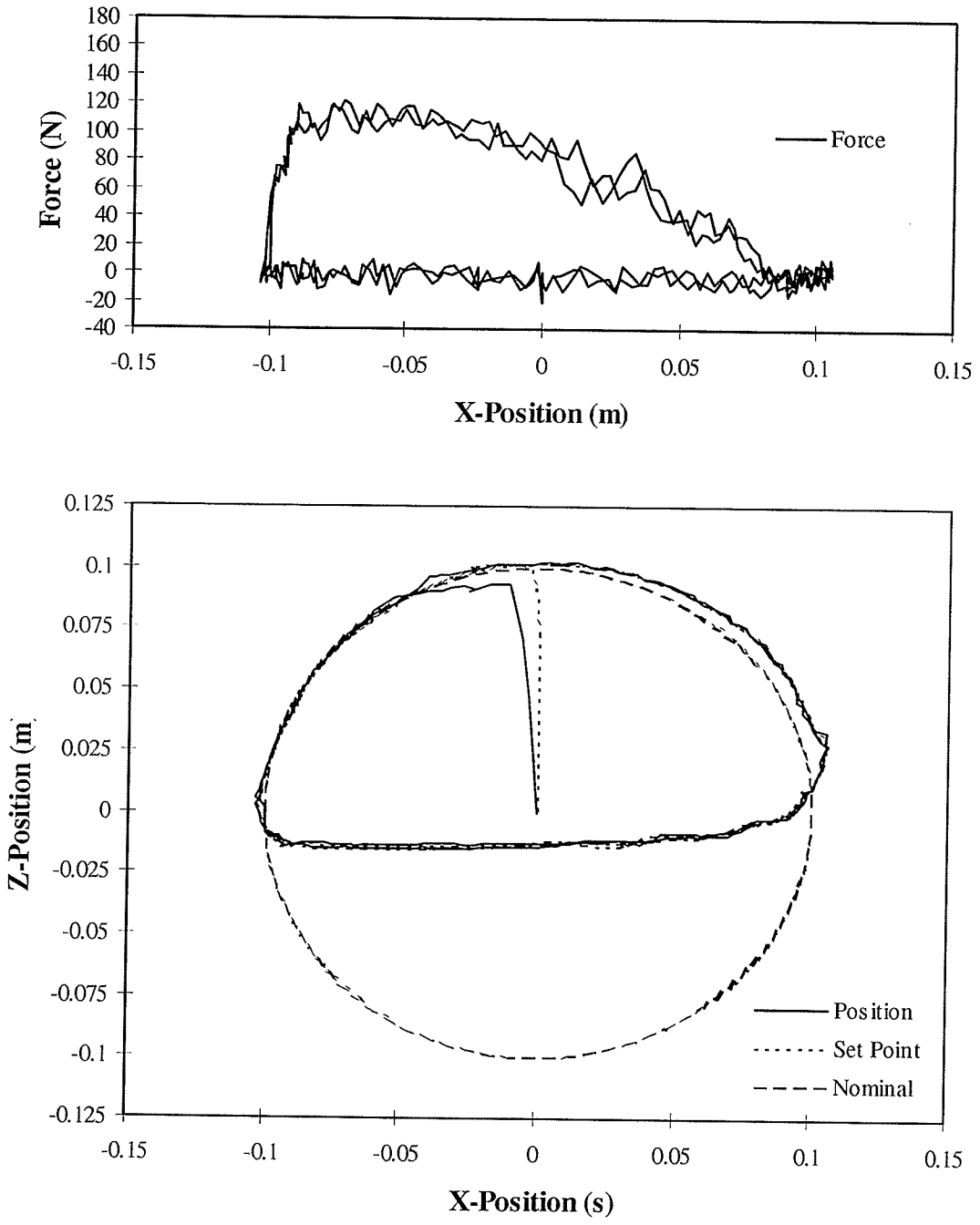


Figure 5-9: **Impact test, K_e = 7200N/m (motion plot, m=50kg, ξ_{net}=1.0, and k=1kN/m).**

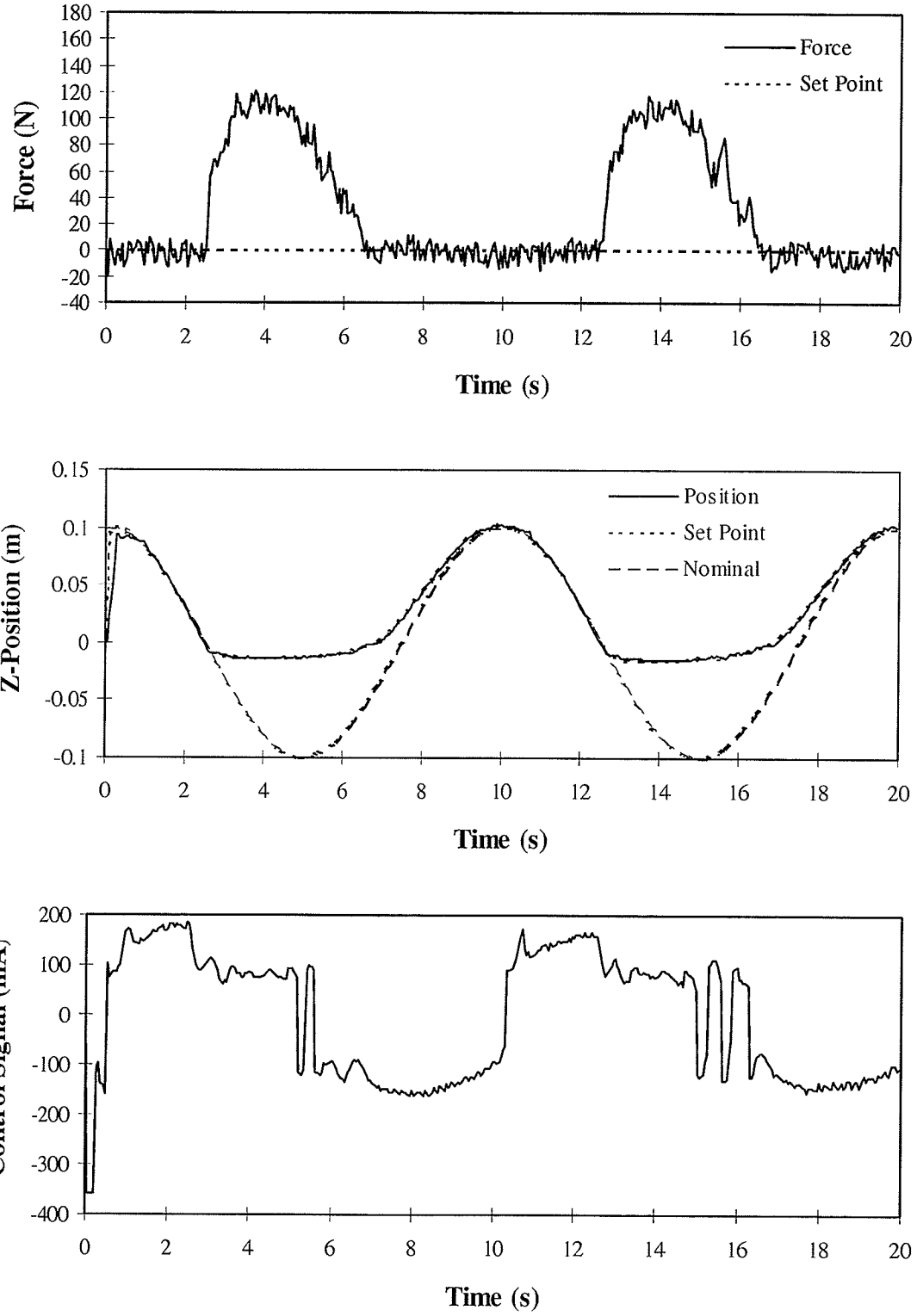
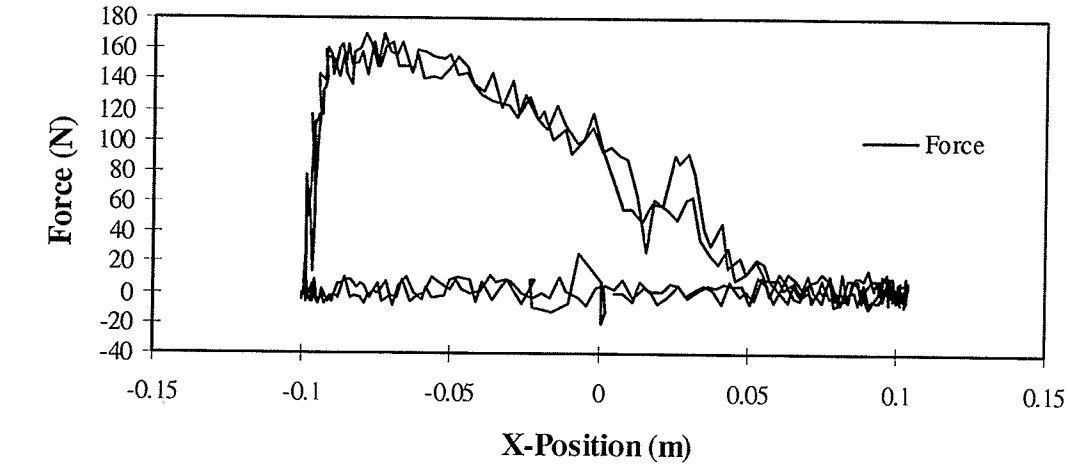


Figure 5-10: Impact test, K_e = 7200N/m (history plot, m=50kg, ξ_{net}=1.0, and k=1kN/m).



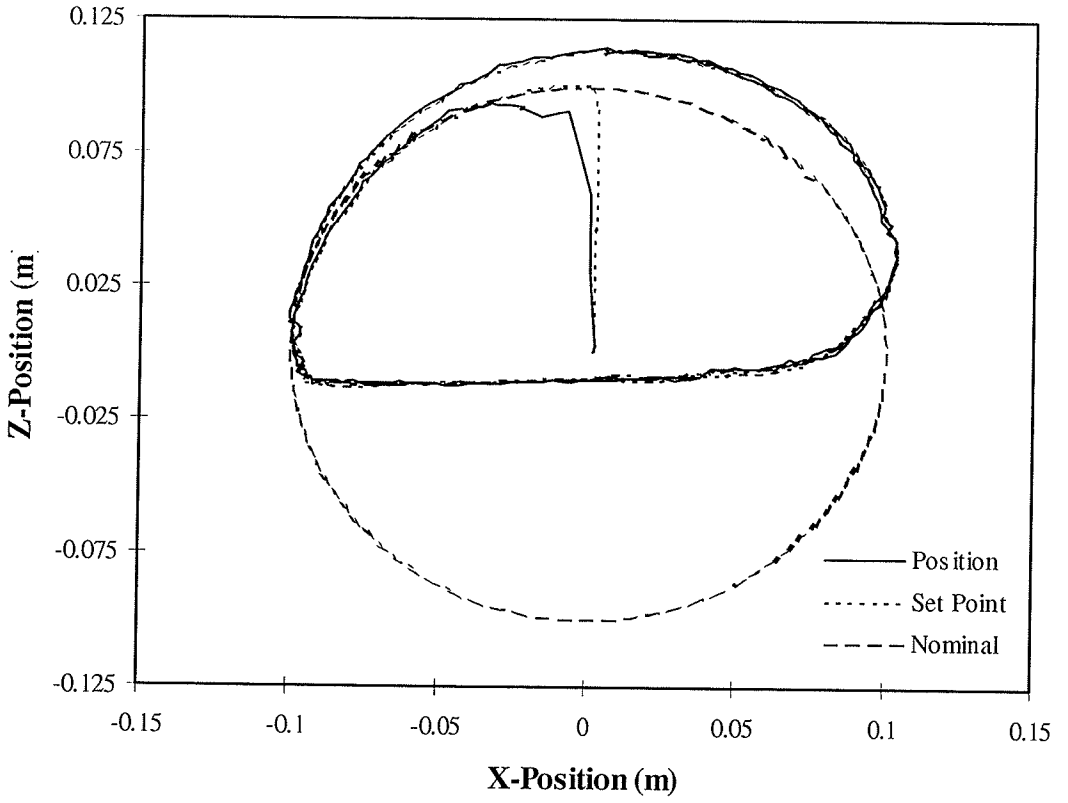
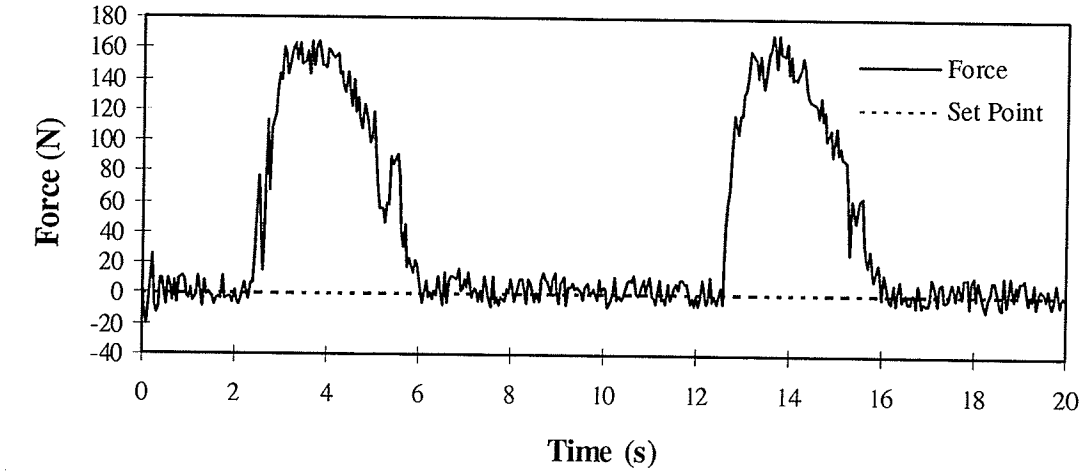
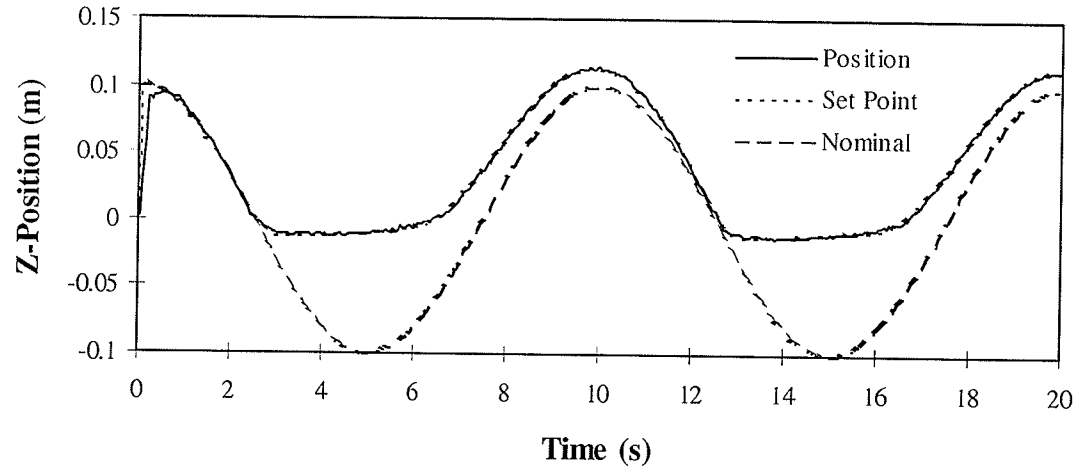


Figure 5-11: Impact test, K_e = 57600N/m (motion plot, m=50kg, ξ_{net}=1.0, and k=1kN/m).





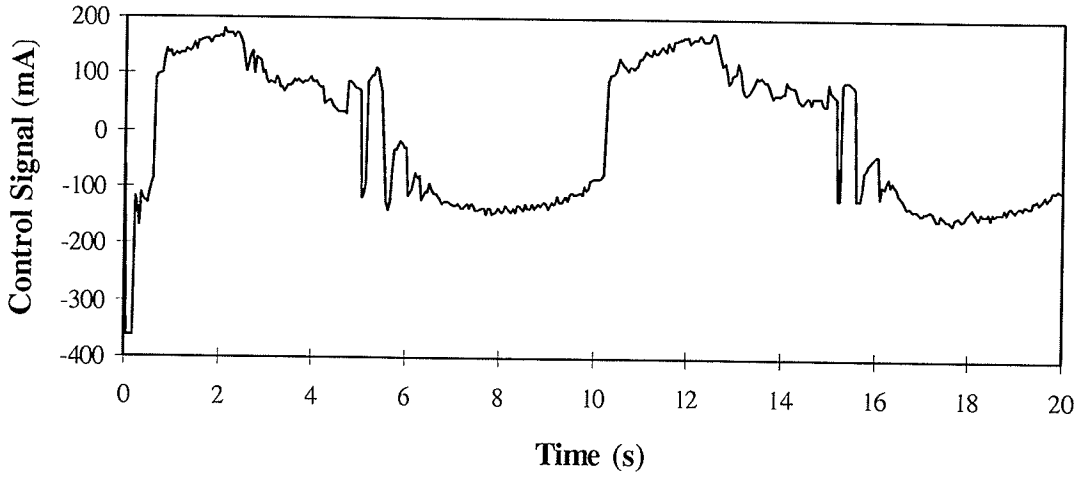


Figure 5-12: Impact test, K_e = 57600N/m (history plot, m=50kg, ξ_{net}=1.0, and k=1kN/m).

5.6 Inertial Obstruction in One-Dimensional Position Tracking

The final set of tests involved placing a heavy crate in the path of the nominal position trajectory. This test demonstrated the controller's ability to minimize forces in the same direction as the position control action, a task that is fundamentally contrary to the basis of hybrid control. An example of an application similar to this task is in controlling a sawblade, where forces must be controller to reduce the chance of binding in sawmill operation.

The target impedance was chosen for this trial in order to best illustrate the controller action. Accordingly, a small target stiffness and fast dynamics were specified: m=500kg, ξ=1.0, and k=5kN/m. The results of the impedance control test were contrasted with those obtained using only position control.

This experiment showed that the impedance controller could balance force and position objectives along the same degree-of-freedom. The position trajectory shown in Figure 5-13 was retarded by about 3cm, which is the required deflection with a mean force of 140N and a target stiffness of 5kN/m. Significant oscillations about the mean force were observed for about 5 seconds in the force readings, indicating a kind of "nudging" action. In fact, upon initially striking the crate, the manipulator actually backed up for another go at it. However, the oscillations ceased once the required position lag was reached.

Comparing the performance of the impedance controller with that of the position controller shows no improvement in response. In fact, the impedance controller caused repeated impacts, and therefore repeated force spikes, whereas the position controller encountered a single spike at impact. The point which should be taken away from this experiment is that the impedance controller can balance force and position objectives in a single dimension. While this particular test of this ability required first an unalterable impact force to break static friction, followed by another unalterable force to overcome kinetic friction, other tasks, such as pushing a sawblade, involve forces that depend on the rate of attack. For tasks such as these, the ability to back off of the nominal position

trajectory that has been demonstrated here would translate into reduced forces of interaction, not oscillation.

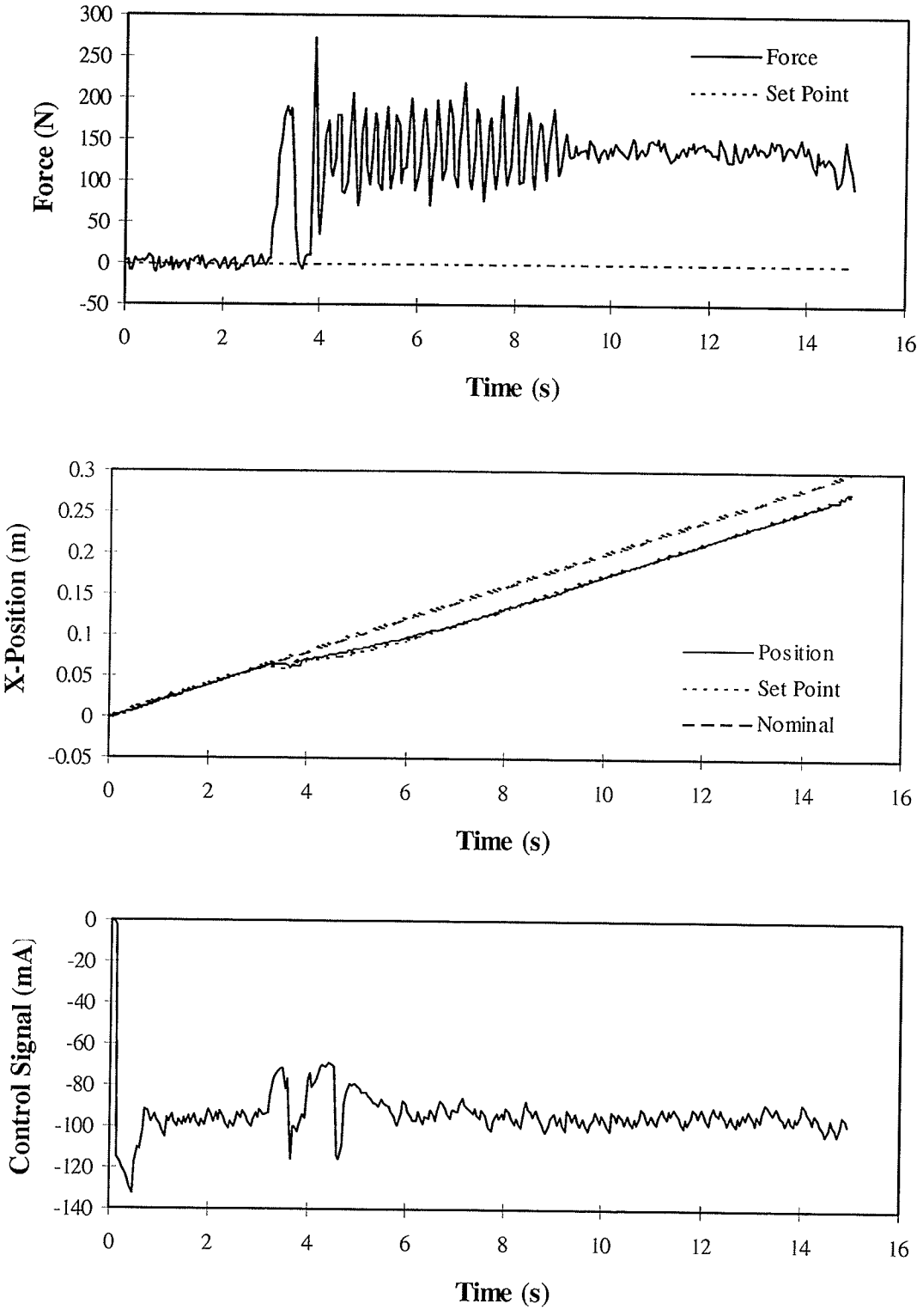


Figure 5-13: Inertial obstruction, impedance control (m=500kg, ξ=1.0, k=5kN/m).

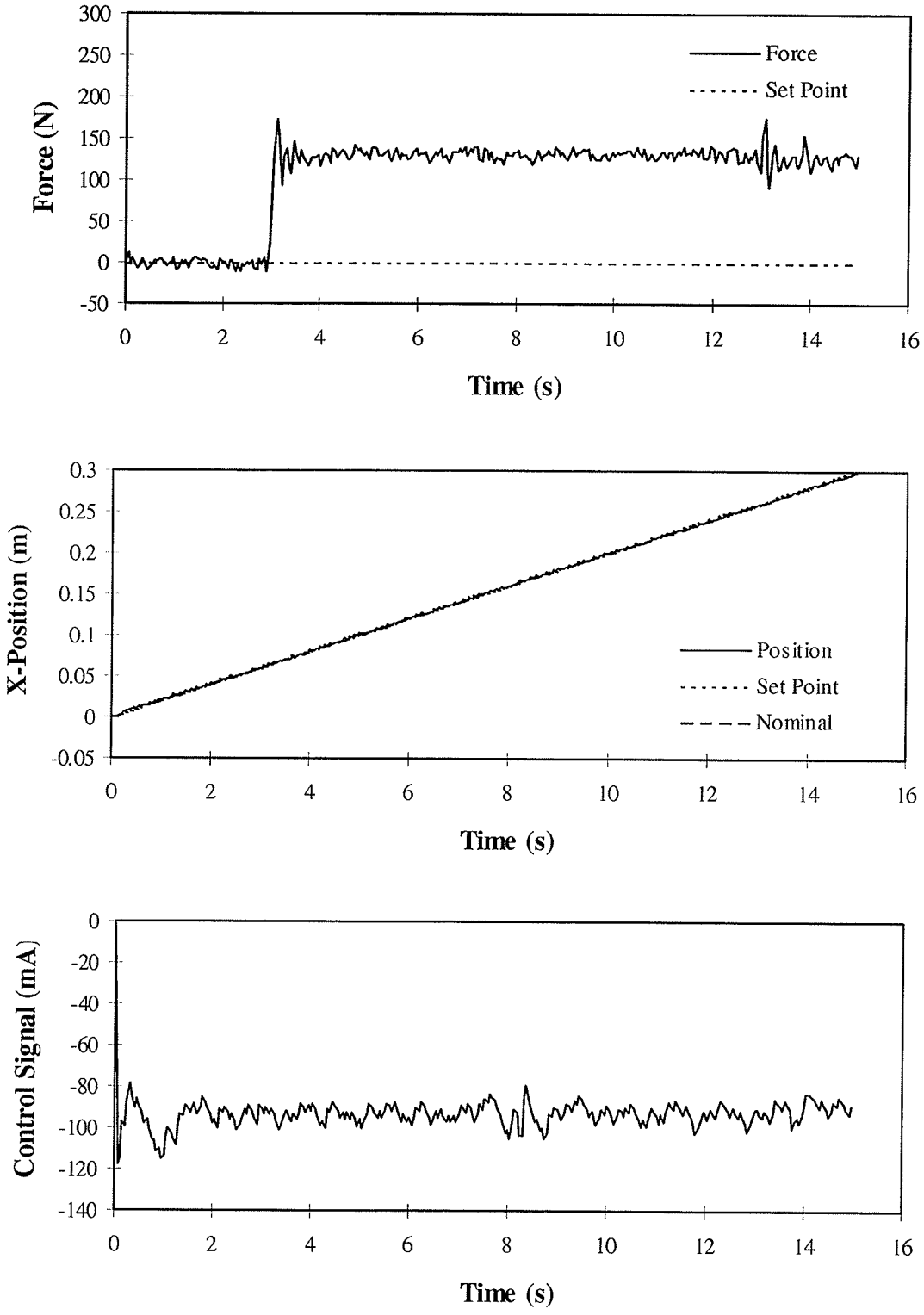


Figure 5-14: Inertial obstruction, position control (no force feedback).

5.7 Summary

Position based impedance control was implemented on an existing, highly non-ideal industrial hydraulic manipulator. The controller was tested in free space, in static and dynamic force control applications, and in applications involving trajectory tracking in environments with highly discontinuous admittances. Tests involving environmental contact were repeated at two values of environmental stiffness: 7kN/m and 53kN/m. In all trials and environments, the controller successfully replaced the actual manipulator dynamics with those of the target impedance. The exception was when contact forces exceeded 200N, in which case the NPI position controller was unable to produce the correct control action.

6. Force Control Equivalence

A relationship between the target impedance parameters and the effective force control gains that exists when using a computed torque formulation of impedance control has been derived in a number of sources (for example, see Hogan, 1987; Volpe and Khosla 1993b). The relationship has been experimentally validated and is widely accepted. Insights gained from this relationship have been shown to have application in stabilizing impact. It will be shown below, however, that this relationship is specific to a computed torque based formulation of impedance control. The relationships that hold for position based formulations of impedance control will then be derived.

This section is organized as follows: first, the proof of the existing relationship is presented and discussed. Secondly, the corresponding relationship is derived for the proposed position based formulation of impedance control. The meaning and consequences of this relationship are discussed. Lastly, a third relationship is derived for an alternate formulation of position based impedance control.

6.1 The Relationship of Torque Based Impedance Control to Force Control

One of the most general proofs of the relationship under question was presented by Volpe and Khosla (1993b). The proof is presented here for purposes of illustration. The equations describing the control law are:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) + \mathbf{K}(\mathbf{x} - \mathbf{x}_d) = -\mathbf{f}$$

$$\tau = \mathbf{D}\ddot{\mathbf{q}}_d + \mathbf{h} + \mathbf{J}^T\mathbf{f}$$

where $\mathbf{x}$ and $\mathbf{x}_d$ are vectors describing the actual and desired Cartesian positions of the end-effector, $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ are the target impedance matrices, $\mathbf{f}$ is the measured force, τ and $\mathbf{q}_d$ are the generalized joint torques and desired joint positions, $\mathbf{D}$ is the joint space inertia matrix, $\mathbf{h}$ is the vector of centripetal, coriolis, gravimetric and frictional joint

torques, and **J** is the manipulator Jacobian. Solving for the desired accelerations and converting joint accelerations to Cartesian accelerations gives the following relations:

$$\tau = \mathbf{J}^T\left[\mathbf{P}(\mathbf{f}_d - \mathbf{f}) + \mathbf{f} - \mathbf{K}_v\dot{\mathbf{x}}\right] + \mathbf{h}^*$$
$$\mathbf{P} = \Lambda\mathbf{M}^{-1}$$
$$\Lambda = \mathbf{J}^{-T}\mathbf{D}\mathbf{J}^{-1}$$
$$\mathbf{f}_d = \mathbf{C}\dot{\mathbf{x}}_d + \mathbf{K}(\mathbf{x}_d - \mathbf{x})$$
$$\mathbf{K}_v = \mathbf{P}\mathbf{C}$$
$$\mathbf{h}^* = \mathbf{h} + \mathbf{J}^T\mathbf{L}\dot{\mathbf{J}}\dot{\mathbf{q}}$$

In converting the desired accelerations from Cartesian space to joint space, the manipulator inertia matrix **D** was converted to an equivalent Cartesian inertia matrix Λ. Neglecting rate feedback and nonlinear force terms, the control torques are simply given by

$$\tau = \mathbf{J}^T[(\mathbf{P} - \mathbf{I})(\mathbf{f}_d - \mathbf{f}) + \mathbf{f}_d)]$$

Therefore, the control law is equivalent to proportional explicit force control with feedforward. The equivalent force control gain is then:

$$\mathbf{K}_P^{FC} = \Lambda\mathbf{M}^{-1} - \mathbf{I}$$

where **I** is the identity matrix. This simple relation shows that, for a torque based impedance controller operating against a stiff environment with negligible sensor dynamics, the control action is equivalent to proportional explicit force control. The equivalent control gain is proportional to the ratio of the actual inertia to the target impedance inertia. This relationship motivated experiments that showed stability using proportional force gains as low as -1 (Volpe & Khosla, 1993b). This can equivalently be thought of as increasing the target inertia to the point where the desired accelerations caused by impact forces are very small.

The above derivation makes a number of assumptions, such as a stiff environment and negligible sensor dynamics. These assumptions allow the lack of consideration of the **Kx** term in the expression for $\mathbf{f}_d$. The end-effector position is assumed to be constant in the presence of a stiff environment. For static force control, this condition requires $\mathbf{f}_d$ =

$\mathbf{K}\mathbf{x}_d$, instead of the correct expression $\mathbf{f}_d = \mathbf{K}_{tot}\mathbf{x}_d$, where $\mathbf{K}_{tot} = \mathbf{K}(\mathbf{K}+\mathbf{K}_{eff})^{-1}\mathbf{K}_{eff}$ and $\mathbf{K}_{eff} = \mathbf{K}_s(\mathbf{K}_s+\mathbf{K}_e)^{-1}\mathbf{K}_e$. This shows that the relationship is under the further constraint that the target stiffness is much less that that of the combined sensor-environment system.

An alternative derivation of this controller equivalence relation will now be shown, for two reasons. First, the technique used will be that used for subsequent proofs, and it is desirable to show its validity. Secondly, this proof will be less restrictive in its assumptions, showing the effects of appreciable environmental compliance quantitatively. The torque based formulation of impedance control is represented in the following diagram:

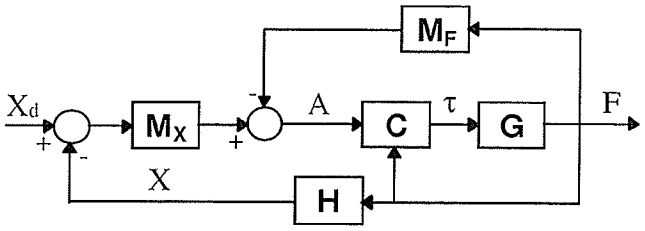


Figure 6-1: Torque based impedance control block diagram.

where **A** is the Laplace transform of the vector of desired Cartesian accelerations. Note that the actuator transfer function **A** is not used in Figure 6-1 because TBIC assumes controllable actuator torques. Assuming further that the nonlinear terms in the dynamic equation can be ignored and that the dynamics of the physical robot-sensor-environment system can be modeled as a linear system as described in Section 3.2, the blocks in Figure 6-1 can be filled with the following transfer functions:

$$\mathbf{M}_X = \mathbf{M}^{-1}(\mathbf{C}s + \mathbf{K})$$
$$\mathbf{M}_F = \mathbf{M}^{-1}$$
$$\mathbf{C} = \mathbf{J}^T(\Lambda\mathbf{A} + \mathbf{F})$$
$$\mathbf{G} = \mathbf{R}_F$$
$$\mathbf{H} = \mathbf{R}_X\mathbf{R}_F^{-1}$$

where $\mathbf{R_F}$ and $\mathbf{R_X}$ are as defined in Section 3.2.1. The control torque generated by this control scheme is

$$\begin{aligned}\tau &= \mathbf{J}^T\Lambda\left[\mathbf{M_X}(\mathbf{X_d}-\mathbf{X})-\mathbf{M_F}\,\mathbf{F}\right]+\mathbf{J}^T\mathbf{F}\\ &= \mathbf{J}^T\left\{\Lambda\left[\mathbf{M_X}\,\mathbf{X_d}-(\mathbf{M_X}\mathbf{H}+\mathbf{M_F})\mathbf{F}\right]+\mathbf{F}\right\}\\ &= \mathbf{J}^T\left\{\left[\Lambda(\mathbf{M_X}\mathbf{H}+\mathbf{M_F})-\mathbf{I}\right](\mathbf{F_d}-\mathbf{F})+\mathbf{F_d}\right\}\end{aligned}$$

where $\mathbf{F_d} = (\mathbf{M_X}\mathbf{H}+\mathbf{M_F})^{-1}\mathbf{M_X}\mathbf{X_d}$. Therefore, in the static limit, the control scheme is equivalent to proportional gain explicit force control with gain matrix $\mathbf{P} = \Lambda(\mathbf{M_X}\mathbf{H} + \mathbf{M_F}) - \mathbf{I}$. In the limiting case as the environment and sensor become stiff, $\mathbf{H}\rightarrow 0$ so $\mathbf{P}\rightarrow\Lambda\mathbf{M_F} - \mathbf{I}$, as derived previously. Note, however, that the dependence of $\mathbf{F_d}$ and $\mathbf{P}$ on the environmental and sensor parameters is now shown. The static component of $\mathbf{F_d}$ now satisfies the desired impedance relationship for any environmental and sensor stiffnesses. Furthermore, $\mathbf{F_d}$ is no longer dependent on position feedback, making the set-point independent of the output variables.

The preceding analysis has shown that a block diagram approach can be used to derive a general relation between the target impedance parameters and the effective explicit force control proportional gain constant for a computed torque based formulation of impedance control. This ability will now be exploited to derive corresponding relations for position based impedance control.

6.2 The Relationship of Position Based Impedance Control to Force Control

6.2.1 Proposed Formulation

The first formulation of position based impedance control that will be considered is that of pure set-point tracking.

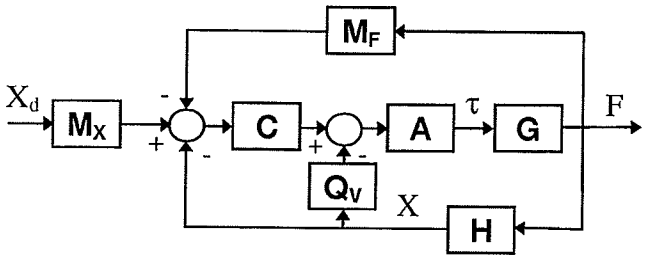


Figure 6-2: Position based impedance control block diagram (proposed formulation).

The transfer functions in Figure 6-2 are the same as those given in above, again with $G = R_F$ and $H = R_X R_F^{-1}$. It is important to note that the functions M_X and M_F are different in the position based formulation than in the torque based formulation. The expression for the control torque is:

$$\begin{aligned}\tau &= J^T A\left[C J^T (M_X X_d - X - M_F F) - Q_v X\right] \\ &= J^T A\left\{C J^T \left[M_X X_d - (H + M_F)F\right] - Q_v H F\right\} \\ &= J^T A\left\{C J^T (H + M_F)(F_d - F) - Q_v H F\right\}\end{aligned}$$

where $F_d = (H+M_F)^{-1} M_X X_d$. In order to determine the equivalent force controller, the implementation of explicit force control on the same system must be analyzed.

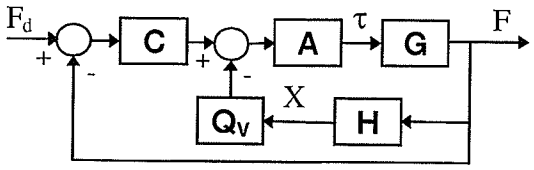


Figure 6-3: Explicit force control block diagram for a hydraulic system.

The control torques generated by an explicit force controller applied to the system are:

$$\begin{aligned}\tau &= J^T A\left\{C(F_d - F) - Q_v X\right\} \\ &= J^T A\left\{C(F_d - F) - Q_v H F\right\}\end{aligned}$$

Equating the control torques of the position based impedance controller and the explicit force controller gives the following relation:

$$\mathbf{C}^{FC} = \mathbf{C}^{PC}\,\mathbf{J}^{T}(\mathbf{H}+\mathbf{M}_F)$$

where the superscripts FC and PC refer to force and position control, respectively. This relationship is very different than that derived for the computed-torque implementation. The underlying behaviour of this equivalent force controller is that of the position controller, modified by a filtering function. Analysis of the static component reveals that, using a proportional position controller, the equivalent explicit force controller exerts the proportional control effort

$$\mathbf{C}^{FC} = \mathbf{J}^{T}\mathbf{K}_p(\mathbf{K}_e^{-1}(\mathbf{K}_e+\mathbf{K}_s)\mathbf{K}_s^{-1} + \mathbf{K}^{-1})$$

In the limiting case of a stiff environment and sensor, this reduces to

$$\mathbf{C}^{FC} = \mathbf{J}^{T}\mathbf{K}_p\mathbf{K}^{-1}$$

Therefore, for position based impedance control, the effective ultimate force control gain is inversely proportional to the target stiffness, as demonstrated in Figure 6-4, whereas the computed torque counterpart was approximately inversely proportional to the target inertia. Figure 6-4 shows how the underlying position controller's behaviour is modified by the filter $\mathbf{J}^{T}(\mathbf{H}+\mathbf{M}_F)$. The plots show the control action of the equivalent explicit force controller given a unit step in the output of the underlying position controller. The steady-state output of the equivalent controller under these conditions is a function of the target stiffness $\mathbf{K}$. The equivalent controller will clearly be unstable for too small a value of $\mathbf{K}$. It will be shown later that the target stiffness must exceed the composite sensor-environment stiffness for stability.

However, basic behaviour of the controller is not limited to proportional control action. Even for a simple second-order linear target impedance, the basic force control action is that of the underlying position controller modified by a second-order filter. This result is interesting in light of the findings of Volpe and Khosla (1993a) wherein it was shown that a second-order filter was the optimal basic form for explicit force control.

The filter can be modified by selection of the target impedance force response. This suggests the possibility of modulating the target impedance transfer functions to cancel the effects of the environment. For example, if $\mathbf{H}$ was known, $\mathbf{M_F}$ could simply be set to $\mathbf{I} - \mathbf{H}$, resulting in $\mathbf{C}^{FC} = \mathbf{C}^{PC}\mathbf{J}^T$. More practically, $\mathbf{M_F}$ could be selected to both cancel $\mathbf{H}$ and obtain the specific control action desired. Corresponding modifications in $\mathbf{M_X}$ could be made to simultaneously retain control over the set-point $\mathbf{F_d}$, since it also depends on the factor $\mathbf{H} + \mathbf{M_F}$. A successful implementation of such a control scheme would be very useful since simultaneous modification of $\mathbf{M_X}$ would prevent the specified force control behaviour from interfering with position control operations.

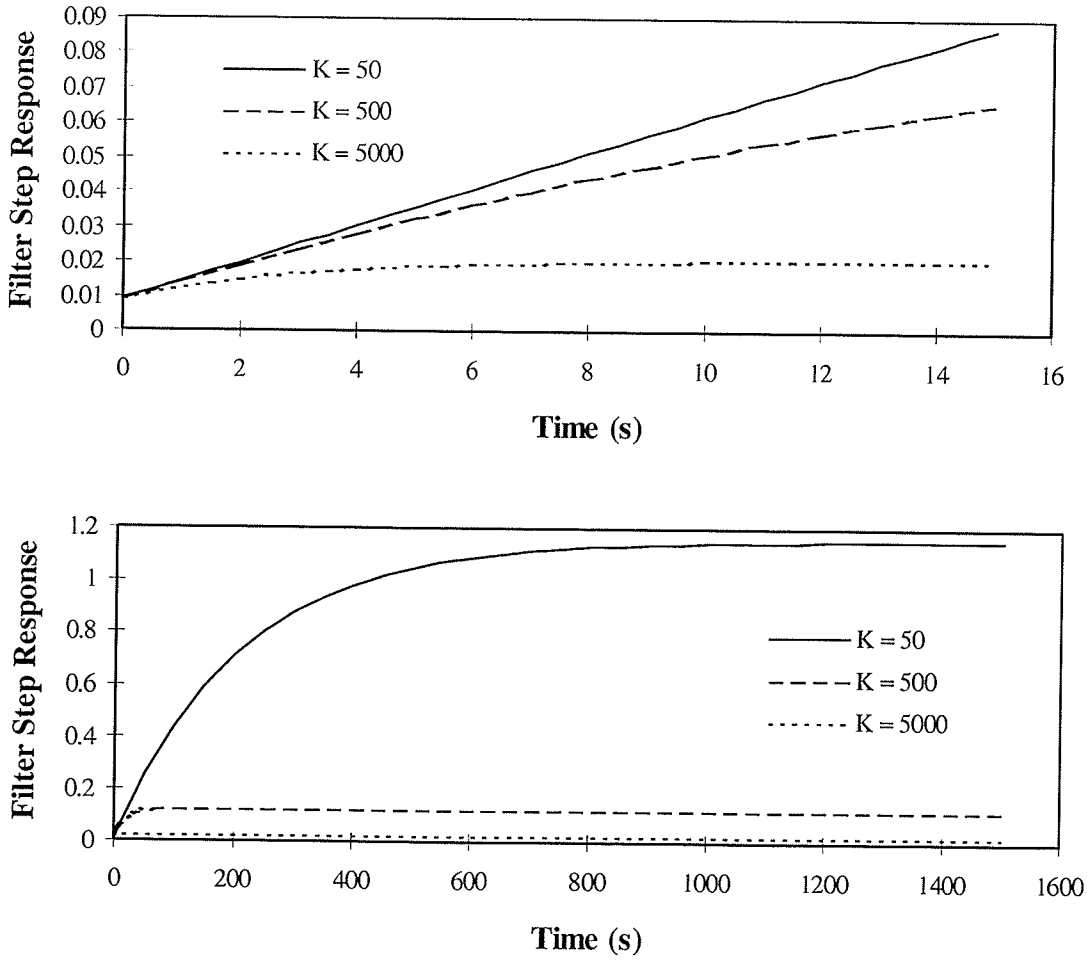


Figure 6-4: Equivalent explicit force controller filter for m=500kg, ξ_{net}=3.0, and various stiffnesses.

6.2.2 Alternate Formulation

There are a number of alternate formulations of PBIC consistent with the general block diagram in Figure 3-4. This possibility arises from the form of the desired impedance relation derived from the system shown in Figure 3-6, written here in matrix form:

$$\mathbf{M}\ddot{\mathbf{x}}_s + \mathbf{C}(\dot{\mathbf{x}}_s - \dot{\mathbf{x}}_d) + \mathbf{K}(\mathbf{x}_s - \mathbf{x}_d) = \mathbf{f}_d - \mathbf{f}$$

In the implementation described previously, the above differential equation was solved for the modified position set-point x_s. However, the acutal position x could be substituted for x_s, or the corresponding derivatives, so long as x_s can still be found. For example, substituting both derivatives and solving for x_s directly can be done as follows:

$$\begin{aligned}
\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) + \mathbf{K}(\mathbf{x}_s - \mathbf{x}_d) &= \mathbf{f}_d - \mathbf{f} \\
\mathbf{K}\mathbf{X}_s + (\mathbf{M}s^2 + \mathbf{C}s)\mathbf{X} - (\mathbf{C}s + \mathbf{K})\mathbf{X}_d &= \mathbf{F}_d - \mathbf{F} \\
\mathbf{X}_s &= \mathbf{K}^{-1}(\mathbf{C}s + \mathbf{K})\mathbf{X}_d - \mathbf{K}^{-1}(\mathbf{M}s^2 + \mathbf{C}s)\mathbf{X} + \mathbf{K}^{-1}(\mathbf{F}_d - \mathbf{F}) \\
\mathbf{X}_s &= \mathbf{M}_{\mathbf{X}_d}^{(a)}\mathbf{X}_d - \mathbf{M}_{\mathbf{X}}^{(a)}\mathbf{X} + \mathbf{M}_{\mathbf{F}}^{(a)}(\mathbf{F}_d - \mathbf{F})
\end{aligned}$$

where the superscripts denote transfer functions for alternate formulation "a". This implementation has the two highly undesirable features of acceleration feedback and unfiltered force feedback. Another, more plausible example is worked out below:

$$\begin{aligned}
\mathbf{M}\ddot{\mathbf{x}}_s + \mathbf{C}(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) + \mathbf{K}(\mathbf{x} - \mathbf{x}_d) &= \mathbf{f}_d - \mathbf{f} \\
\mathbf{M}s^2\mathbf{X}_s + (\mathbf{C}s + \mathbf{K})(\mathbf{X} - \mathbf{X}_d) &= \mathbf{F}_d - \mathbf{F} \\
\mathbf{X}_s &= s^{-2}\mathbf{M}^{-1}(\mathbf{C}s + \mathbf{K})(\mathbf{X}_d - \mathbf{X}) + s^{-2}\mathbf{M}^{-1}(\mathbf{F}_d - \mathbf{F}) \\
\mathbf{X}_s &= \mathbf{M}_{\mathbf{X}}^{(b)}(\mathbf{X}_d - \mathbf{X}) + \mathbf{M}_{\mathbf{F}}^{(b)}(\mathbf{F}_d - \mathbf{F})
\end{aligned}$$

This formulation of the impedance block is strongly related to that used in torque based impedance control (TBIC). The value of X_s obtained here is merely the result of integrating the accelerations calculated for TBIC. This formulation is examined now to determine whether or not the formulation of PBIC affects the form of the equivalent force controller.

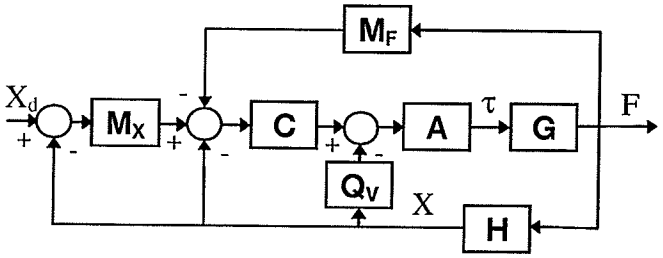


Figure 6-5: Position based impedance control block diagram (alternate formulation).

In Figure 6-5, the impedance transfer functions are again different. In this case, they result from solving the state equation for the position given by integrating the accelerations. For example, using a second-order system, these transfer functions are derived as follows:

$$\begin{aligned}
\mathbf{M}\ddot{\mathbf{x}}_s + \mathbf{C}(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) + \mathbf{K}(\mathbf{x} - \mathbf{x}_d) &= -\mathbf{f} \\
\mathbf{M}s^2\mathbf{X}_s + (\mathbf{C}s + \mathbf{K})(\mathbf{X} - \mathbf{X}_d) &= -\mathbf{F} \\
\mathbf{X}_s &= s^{-2}\mathbf{M}^{-1}(\mathbf{C}s + \mathbf{K})(\mathbf{X}_d - \mathbf{X}) - s^{-2}\mathbf{M}^{-1}\mathbf{F} \\
\mathbf{X}_s &= \mathbf{M}_X(\mathbf{X}_d - \mathbf{X}) - \mathbf{M}_F\mathbf{F}
\end{aligned}$$

Note that these transfer functions are the same as those used in the computed torque based formulation, with a factor of s^{-2} to convert from accelerations to positions. Regardless of the form of these transfer functions, the control torques are found as usual:

$$\begin{aligned}
\tau &= \mathbf{J}^T\mathbf{A}\{\mathbf{C}\,\mathbf{J}^T[\mathbf{M}_X(\mathbf{X}_d - \mathbf{X}) - \mathbf{X} - \mathbf{M}_F\,\mathbf{F}] - \mathbf{Q}_v\,\mathbf{X}\} \\
&= \mathbf{J}^T\mathbf{A}\{\mathbf{C}\,\mathbf{J}^T[\mathbf{M}_X\,\mathbf{X}_d - ((\mathbf{M}_X + \mathbf{I})\mathbf{H} + \mathbf{M}_F)\mathbf{F}] - \mathbf{Q}_v\mathbf{H}\mathbf{F}\} \\
&= \mathbf{J}^T\mathbf{A}\{\mathbf{C}\,\mathbf{J}^T((\mathbf{M}_X + \mathbf{I})\mathbf{H} + \mathbf{M}_F)(\mathbf{F}_d - \mathbf{F}) - \mathbf{Q}_v\mathbf{H}\mathbf{F}\}
\end{aligned}$$

where $\mathbf{F}_d = (\mathbf{M}_F + (\mathbf{M}_X + \mathbf{I})\mathbf{H})^{-1}\mathbf{M}_X\mathbf{X}_d$. The equivalent explicit force controller is then:

$$\mathbf{C}^{FC} = \mathbf{C}^{PC}\,\mathbf{J}^T[(\mathbf{M}_X + \mathbf{I})\mathbf{H} + \mathbf{M}_F]$$

Filling in the values for the model transfer functions $\mathbf{M}_X$ and $\mathbf{M}_F$ shows that the same expressions for $\mathbf{F}_d$ are obtained in both formulations.

$$\mathbf{F}_d^{(1)} = (\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K} + \mathbf{K}_e)^{-1}\,\mathbf{M}_X\,\mathbf{X}_d = \mathbf{F}_d^{(2)}$$

However, different expressions are obtained for the equivalent explicit force controllers:

$$\mathbf{C}^{FC(1)} = \mathbf{C}^{PC}\,\mathbf{J}^{T}\mathbf{K}_e{}^{-1}(\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K})^{-1}(\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K} + \mathbf{K}_e)$$
$$\mathbf{C}^{FC(2)} = \mathbf{C}^{PC}\,\mathbf{J}^{T}\mathbf{K}_e{}^{-1}\mathbf{M}^{-1}s^{-2}(\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K} + \mathbf{K}_e)$$

Evidently, the filter for the second formulation's equivalent controller has no finite steady-state. This has direct consequences for the stability of this formulation that differ considerably from those of the first formulation. Therefore, the equivalence analysis needs to be conducted for each specific formulation of PBIC to be implemented.

The reason that this alternate formulation, while probably feasible, was not implemented because the motion of the set-point is affected by the current position, as well as the current force. Thus, it is doubly susceptible to nonlinear effects such as stiction. It was deemed preferable to develop a robust position controller than to deal with extra motions in the set-point.

6.3 Experimental Verification

The theoretical results were tested using the apparatus described in Section 4. The supports on the aluminum plate were set to the widest spacing, making the net environment-plus-sensor stiffness equal to 6200N/m (see Appendices B & E). The end-effector was placed in contact with the plate for the start of the test. In order to maintain consistency with the linearized analysis, however, a linear PI position controller was used, rather than the fully modified NPI controller.

The test was of the response of each controller to a step change in force set-point. A regulation task was chosen in order to test the steady-state system performance. Two control schemes were compared: the proposed position based impedance control formulation (Figure 6-2) and its theoretically equivalent force controller (Figure 6-3). The underlying position controller was implemented as a simple PI controller, the only modification of which was to perform a thresholding operation on the error signal. Only

the second (up/down) link of the Unimate was allowed to move, fixing the third link at one meter extension for a constant manipulator Jacobian.

For simplicity, the impedance parameters were chosen so as to test the equivalence relation without requiring the inclusion of environmental or sensor dynamics in the equivalent explicit force controller filtering function. That is, the natural frequency of the target impedance was chosen to be small compared with that of the composite sensor-environment system. To this end, and to mask the high noise level in the force sensor readings (see Section 4), the target mass, m, was set to 500kg. Net system damping, ξ_{net}, was set to 3.0. The target stiffness, k, was set to three different values: 50N/m, 500N/m, and 5000N/m. Stiffness was varied to allow testing of equivalence relations in both the controller action and force set-point modification, since both of these relations depend on stiffness only in the steady-state.

The plots show a very high correlation between the response of the impedance controller and that of the proposed equivalent force controller. Force, position, and even control signal histories compare very closely. The difference observed between the appearance of the control signals is due to the nature of their respective source error signals. The impedance controller signal is relatively steady but moves in steps. This is due to the fact that the position encoder signal is free from noise but is operating in a range where the encoder resolution is visually evident. Likewise, the explicit force controller signal is noisy because the force signal is noisy. However, both controllers take the same actions at the same times, showing their equivalence.

The expression found for the equivalent force set-point is also verified. The equivalent steady-state force set-points required by sharing a 100N load between the model and the environment for model stiffnesses of 50, 500, and 5000N/m are 99N, 92N, and 55N, respectively. These are indeed the approximate steady-state levels attained in both impedance and force control trials. As the model stiffness is increased, it will bear a greater fraction of the virtually-imposed set-point force.

A number of consequences for the use of a position based impedance controller can be seen at this point. First, any force control application of PBIC will require some

form of compensation for the set-point attenuation observed with increasing target stiffness. This compensation could take the form of an artificially increased force set-point. Secondly, the composite model/sensor/environment stiffness has been shown to be the critical factor in determining system stability. A low composite stiffness results in greater steady-state controller gains (see Figure 6-4). This suggests an eventual system destabilization for small composite stiffnesses. Inspection of Figure 6-4 shows that the trial durations were insufficient for the controller signal to reach full strength when a low target stiffness is specified. The system behaviour in longer experiments is tested later in this work (Section 7). Thirdly, the impedance mass has been shown both analytically and experimentally not to be the primary factor affecting force control equivalence, as it is in the case of torque based impedance control. This has direct implications for the selection of target impedances for applications involving environmental impact in particular. For further evidence of the non-reliance of system stability on the selection of target mass, see Section 7.

6.4 Summary

A commonly-used relationship between gains in an impedance controller and its equivalent explicit force controller was shown to be true only for torque based formulations of impedance control. A new relationship was derived for position based impedance control. Experimental results verified this relationship in a static force control application.

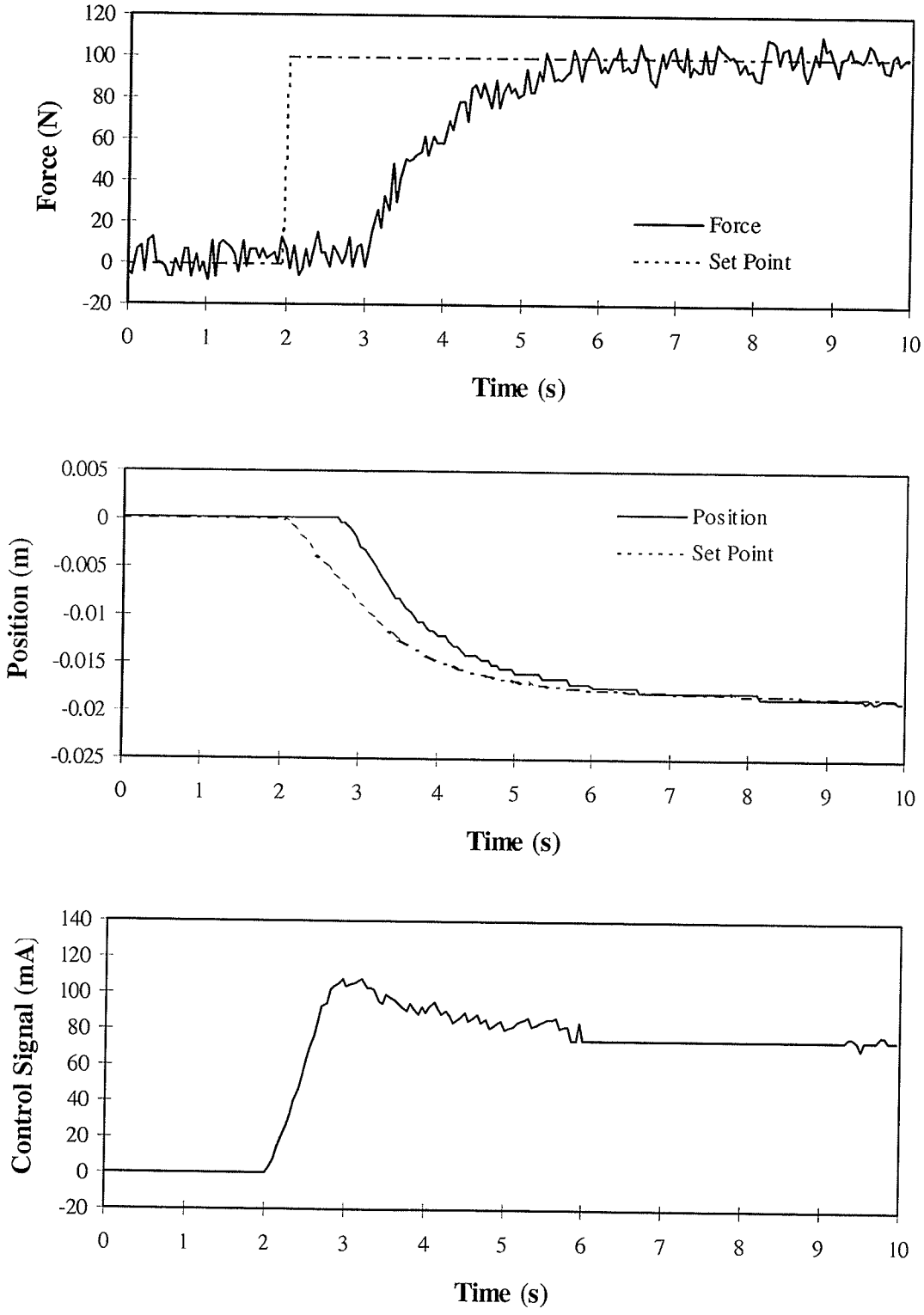
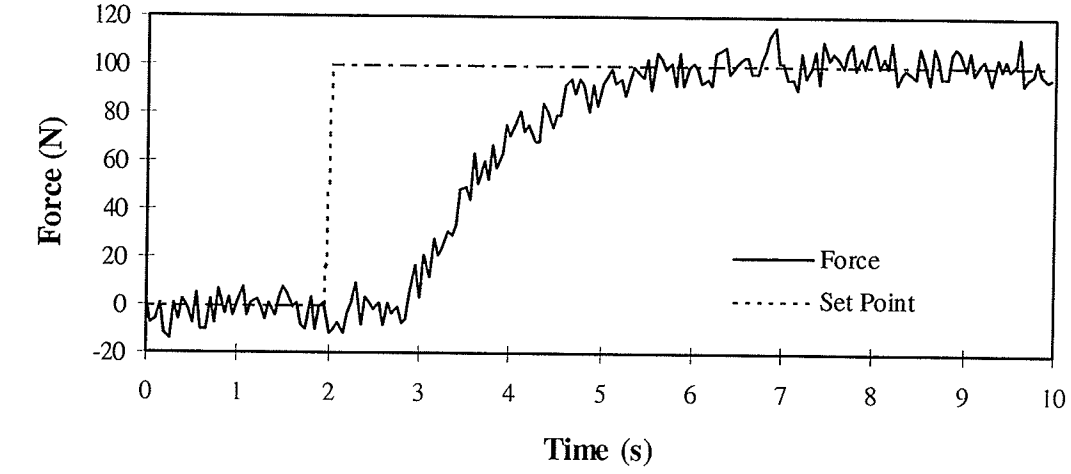
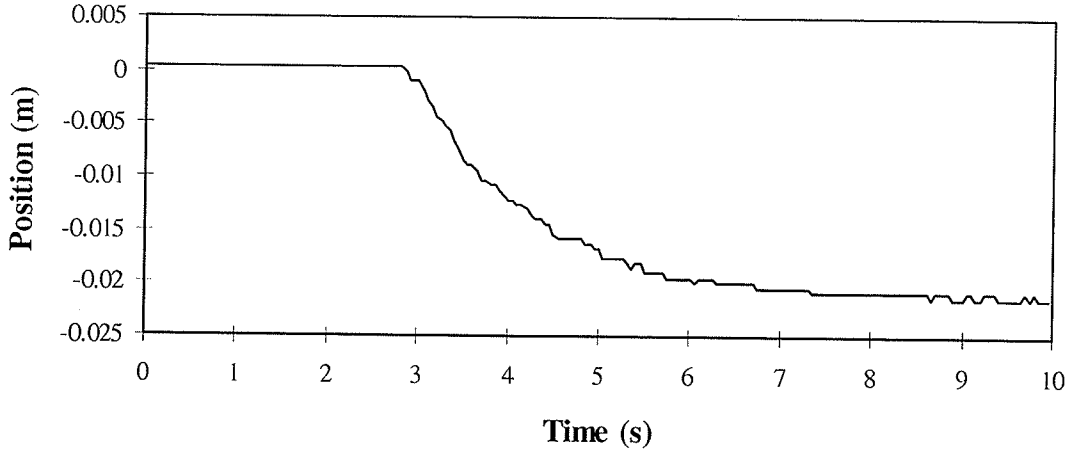


Figure 6-6: Impedance controller, m=500kg, ξ_{net}=3.0, k=50N/m.





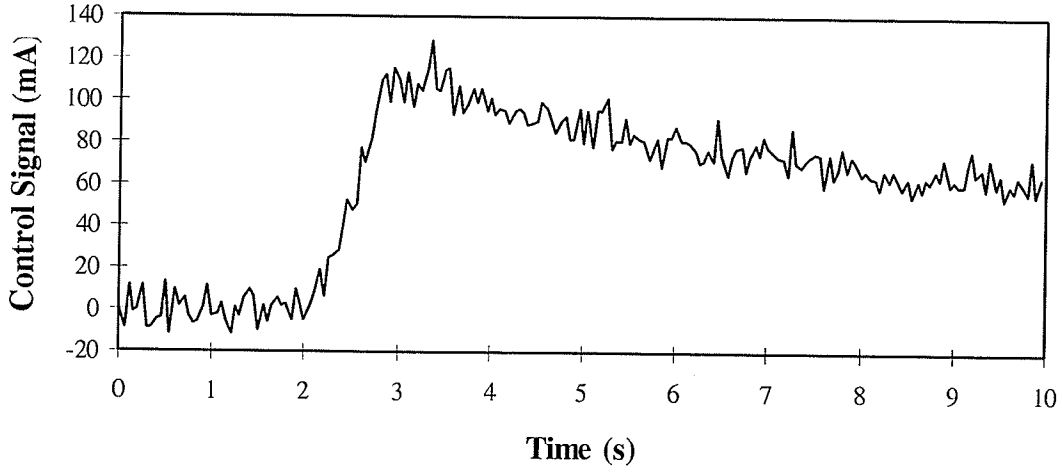


Figure 6-7: Equivalent explicit force controller, m=500kg, ξ_{net}=3.0, k=50N/m.

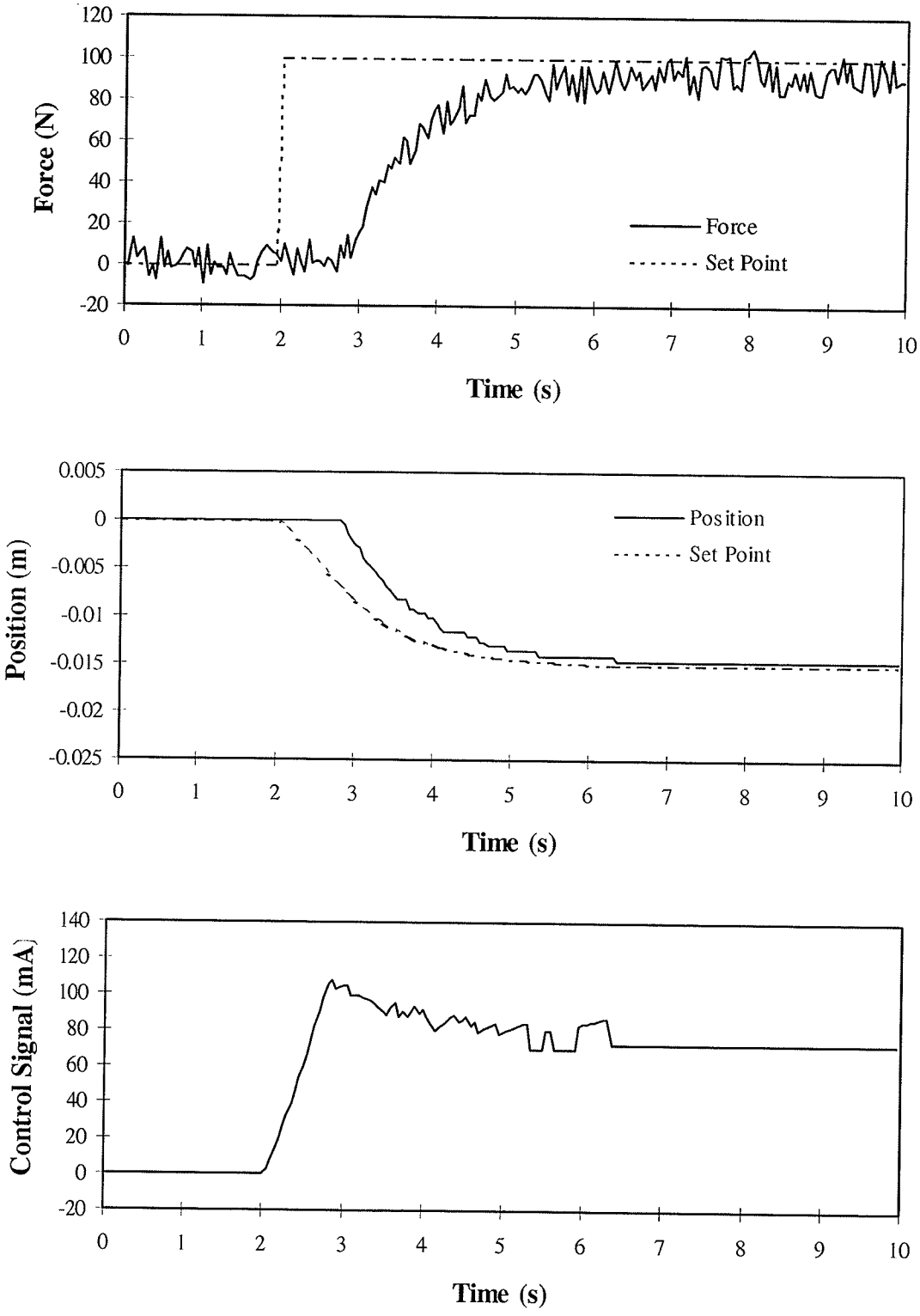


Figure 6-8: Impedance controller, m=500kg, ξ_{net}=3.0, k=500N/m.

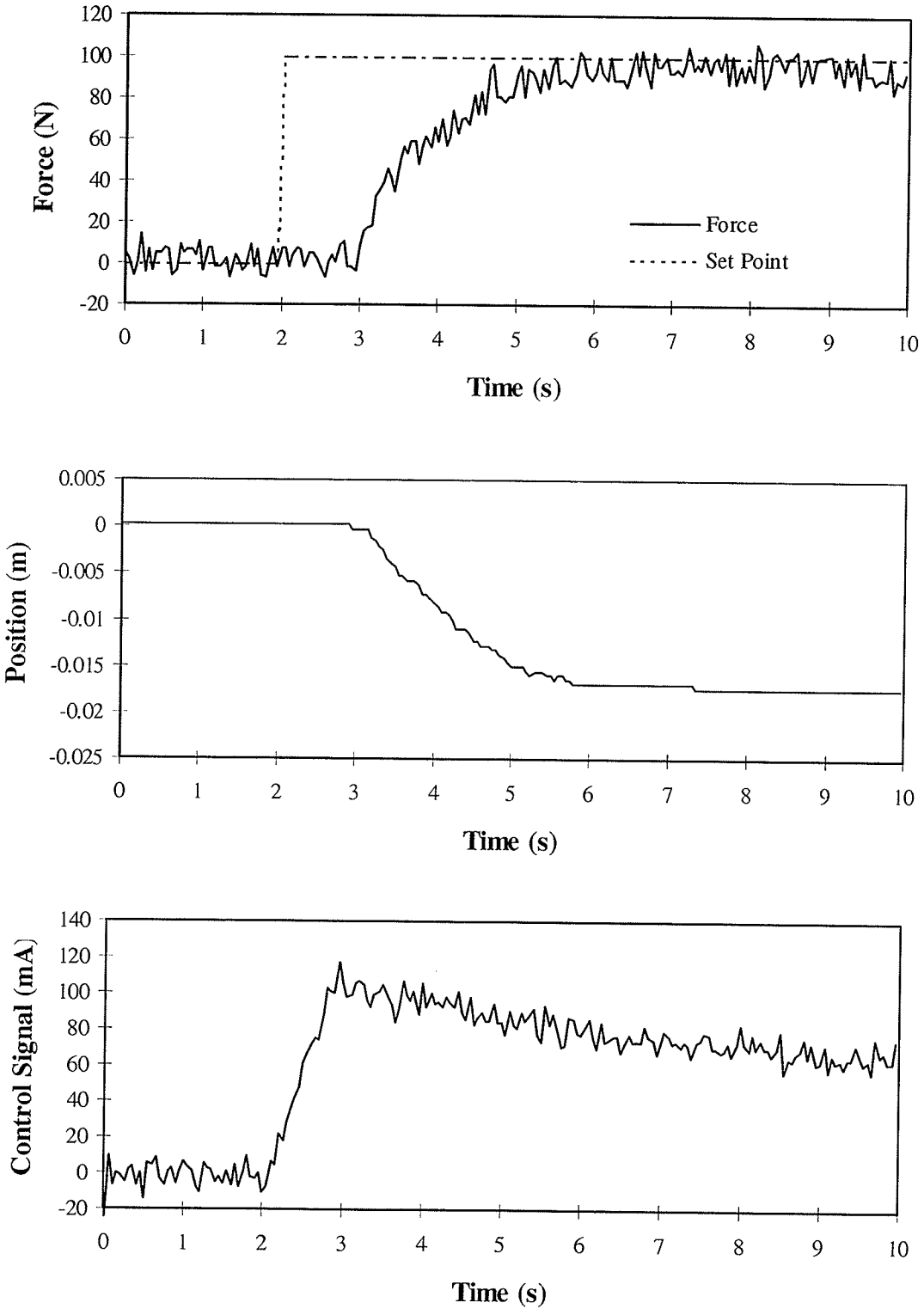


Figure 6-9: Equivalent explicit force controller, m=500kg, ξ_{net}=3.0, k=500N/m.

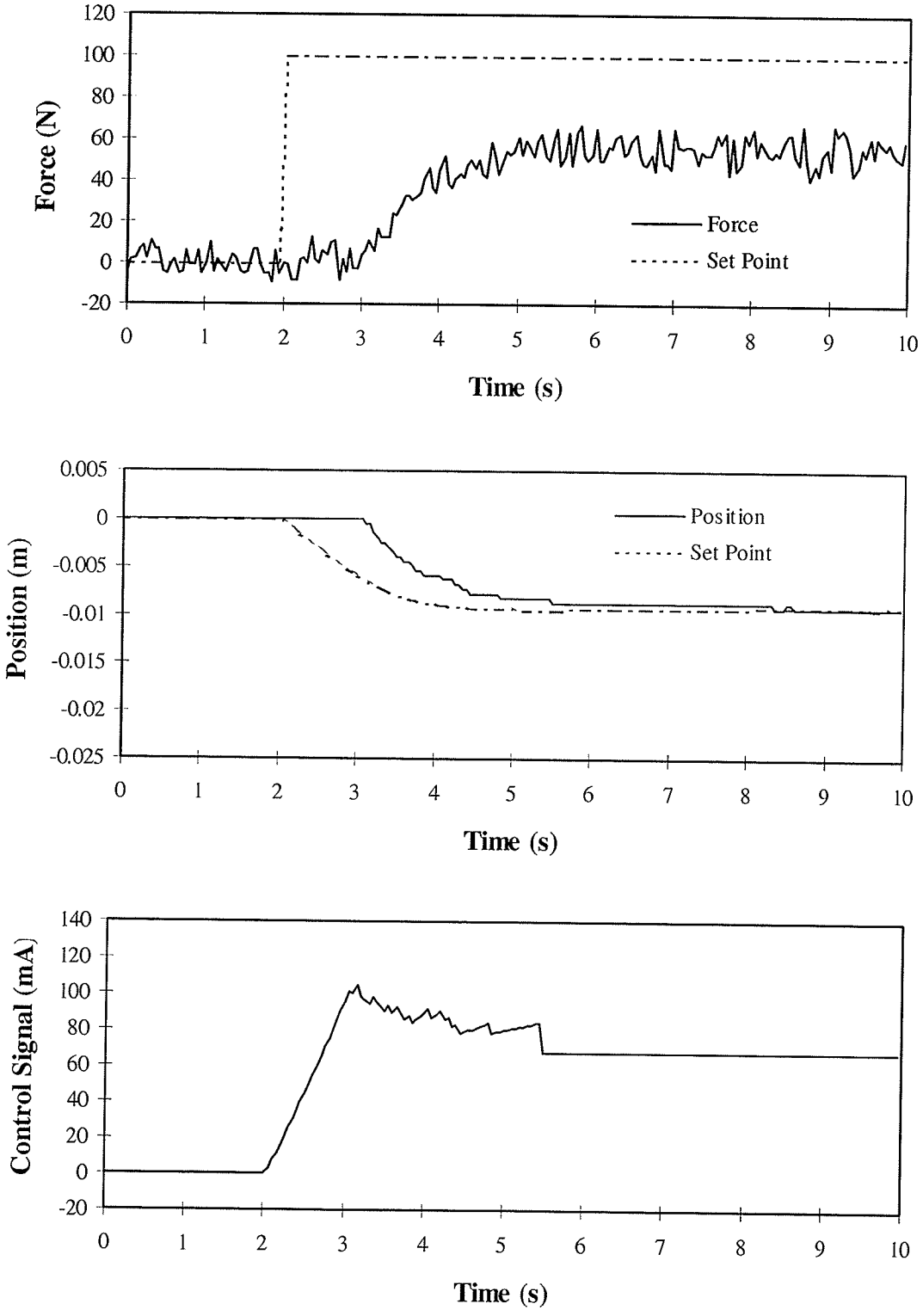


Figure 6-10: Impedance controller, m=500kg, ξ_{net}=3.0, k=5000N/m.

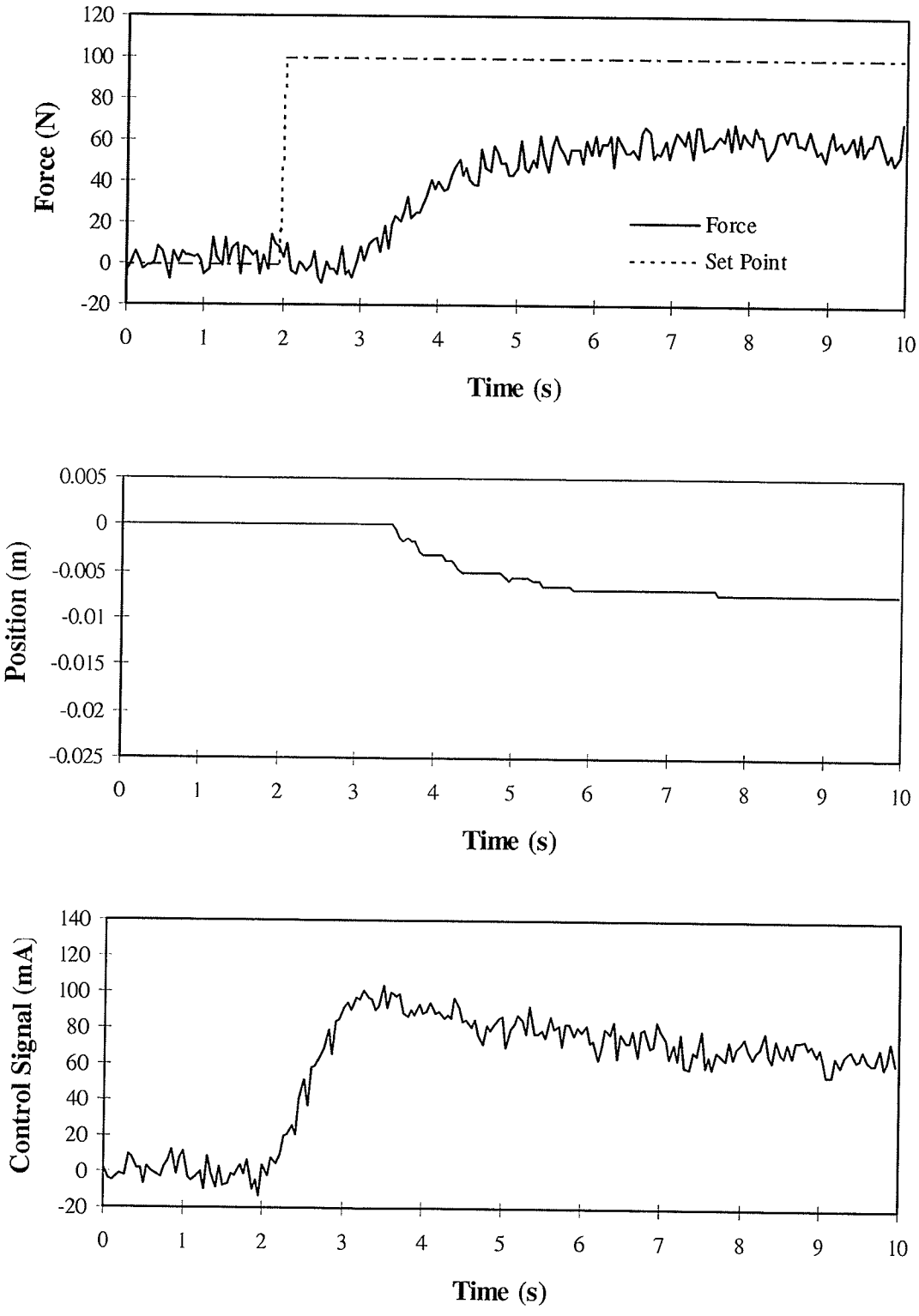


Figure 6-11: Equivalent explicit force controller, m=500kg, ξ_{net}=3.0, k=5000N/m.

7. Limitations of Position Based Impedance Control

7.1 Theoretical Limits of PBIC in a Real System

The analysis presented thus far assumes linearity in the robot, sensor, and the environment. In reality, of course, this is far from the case. The interaction of two nonlinearities, finite encoder resolution and Coulomb friction, will now be examined. Consider a single degree-of-freedom system in continuous contact with the environment controlled using the proposed position based formulation of impedance control. Let us define $\delta\mathbf{q}$ as the vector of the resolutions of the encoders used to measure the generalized joint displacement vector $\mathbf{q}$. The corresponding Cartesian resolution δx is then given by

$$\delta x = \mathbf{J}\delta\mathbf{q}$$

where $\mathbf{J}$ is the manipulator Jacobian. The joint positioning accuracy must be limited by $\delta\mathbf{q}$; in fact, if the position controller does not ignore errors of this magnitude, a set-point which does not coincide with an encoder position will result in error integral accumulation and therefore limit cycling. Therefore, the Cartesian position accuracy is limited by $\delta\mathbf{x}$.

Let us continue to designate the Cartesian stiffnesses of the sensor, environment, and target impedance as $\mathbf{K_S}$, $\mathbf{K_E}$, and $\mathbf{K}$, respectively. In the steady-state, the force read by the sensor is $\mathbf{f_s} = \mathbf{K_S}(\mathbf{x}-\mathbf{x_e}) = \mathbf{K_E}\mathbf{x_e}$, where $\mathbf{x}$ and $\mathbf{x_e}$ are the displacements from the robot-sensor and sensor-environment junctions, respectively. Solving for $\mathbf{f_s}$ gives

$$\begin{aligned}\mathbf{f_s} &= \mathbf{K_E}(\mathbf{K_E} + \mathbf{K_S})^{-1}\mathbf{K_S}\mathbf{x} \\ &= \mathbf{K_{EFF}}\mathbf{x}\end{aligned}$$

Then, assuming negligible sensor dynamics, the change in force resulting from a joint displacement of one encoder resolution is

$$\delta\mathbf{f_s} = \mathbf{K_{EFF}}\delta\mathbf{x}$$

The steady-state change in position set-point $\delta\mathbf{x_s}$ due to a change in force sensor readings $\delta\mathbf{f_s}$ required by the target impedance is

$$\delta \mathbf{x}_s = -\mathbf{K}^{-1}\delta \mathbf{f}_s$$

Let us now consider the following scenario: the current manipulator position **x** read by the encoders is separated from the set-point by a distance $\mathbf{e} = \mathbf{A}\delta\mathbf{x}$, where **A** is a diagonal matrix with $1 < |a_{ii}| < 2$. Let us assume that the controller successfully positions the end-effector $\mathbf{B}\delta\mathbf{x}$ closer to the set-point, where **B** is also diagonal with $1 < |\beta_{ii}| < 2$ as well. The remaining error in position calculated is still $(\mathbf{A}\text{-}\mathbf{I})\delta\mathbf{x}$, since the portion of $\mathbf{B}\delta\mathbf{x}$ over $\delta\mathbf{x}$ cannot be measured. The motion results in an increase in force read by the force sensor of $\delta\mathbf{f}_s = \mathbf{K}_{EFF}\mathbf{B}\delta\mathbf{x}$. The reaction required by the target impedance in response to the increased force results in a new error.

$$\begin{aligned}\mathbf{e}' &= \mathbf{e} + \delta\mathbf{x}_s \\ &= (\mathbf{A} - \mathbf{I})\delta\mathbf{x} - \mathbf{K}^{-1}\delta\mathbf{f}_s \\ &= \left\{(\mathbf{A} - \mathbf{I}) - \mathbf{K}^{-1}\mathbf{K}_{EFF}\mathbf{B}\right\}\delta\mathbf{x}\end{aligned}$$

In order to assure that there will be no limit cycling, each component of this new error must be less the corresponding component of the Cartesian resolution $\delta\mathbf{x}$. This leads to the following condition for each component of $\mathbf{e}'$:

$$\begin{aligned}|e'| &< \delta x \\ -\left(\alpha - 1 - k^{-1}k_{eff}\beta\right)\delta x &< \delta x \\ k^{-1}k_{eff}\beta &< \alpha \\ k &> k_{eff}\beta\alpha^{-1}\end{aligned}$$

The lower-case letters are the diagonal entries of the matrices used for the above analysis, with subscripts dropped for clarity. The stiffness matrices were also assumed to be diagonal. The important result of the final equation stems from the fact that, by definition, the factor $\beta\alpha^{-1}$ is between 0.5 and 2. Therefore, given <u>any</u> finite encoder resolution, limit cycles will occur unless the static component of the target impedance is greater than half the composite stiffness of the environment and sensor. Limit cycles may still occur unless the specified stiffness is the double of that encountered in the

environment. Note that a compliant sensor will improve system stability by decreasing the effective environmental stiffness.

In most applications, there is probably a finite positioning accuracy limit which is greater than the encoder resolution. If such is the case, the vector $\delta\mathbf{q}$ in the above analysis can be replaced with $\delta\mathbf{q}_p$, the vector of joint positioning accuracy limits. The resulting conditions for instability are the same, but any limit cycles will be of greater amplitude because of the greater positioning error.

7.2 Experimental Verification

The long term static force control abilities of the impedance controller were tested by extending the duration of the experiments performed in Section 6.3. The plate supports were set to obtain an environmental stiffness of 7200N/m, combining with the sensor to give a net effective stiffness of 6200N/m. Four target stiffnesses were used: 50, 500, 5000, and 50000N/m. To show that damping does not improve stability, the damping constants were set so that the complete robot-sensor-environment system would be overdamped for $k < k_{eff}$, and underdamped for $k > k_{eff}$. The NPI position controller of Section 4.6 was used.

The results showed the predicted behaviour. All trials where the target stiffness was less than the effective environmental stiffness resulted in eventual limit cycles. The source of these limit cycles can be most clearly seen in the position plots, where the set-point moves in response to the change in force resulting from actually reaching the set-point. The amplitude of oscillation was approximately 3mm (see Figure 7-2, for example), corresponding to 0.17° of up/down motion, the previously established minimum tracking error. The onset of these cycles was slowest for the lowest value of stiffness because of the low natural frequency of that system. The frequency of cycling appeared to increase with the higher natural frequencies resulting from increased target stiffness. Note, however, that the oscillations became sporadic in the k=5000N/m trial. This lends further confidence to the theoretical results, since this value of stiffness lies within the predicted 50% to 200% band about k_{eff} where limit cycles may or may not occur. Finally, no limit

cycles were observed for the greatest value of stiffness (see Figure 7-4), which is greater than double the effective environmental stiffness. This stability is observed in spite of the highly underdamped response of the system.

Further tests were performed to refute the possibility that the selection of the target mass has an effect on the ultimate behaviour of the system. Figure 7-6 shows the result of increasing the target mass by a factor of 100 in the low stiffness trial. The increased mass slows the system response, but does not prevent the eventual occurrence of limit cycles. Conversely, a trial using a low mass and high stiffness fails to indicate any likelihood of instability. Clearly, the target stiffness is the only quantity affecting long term stability.

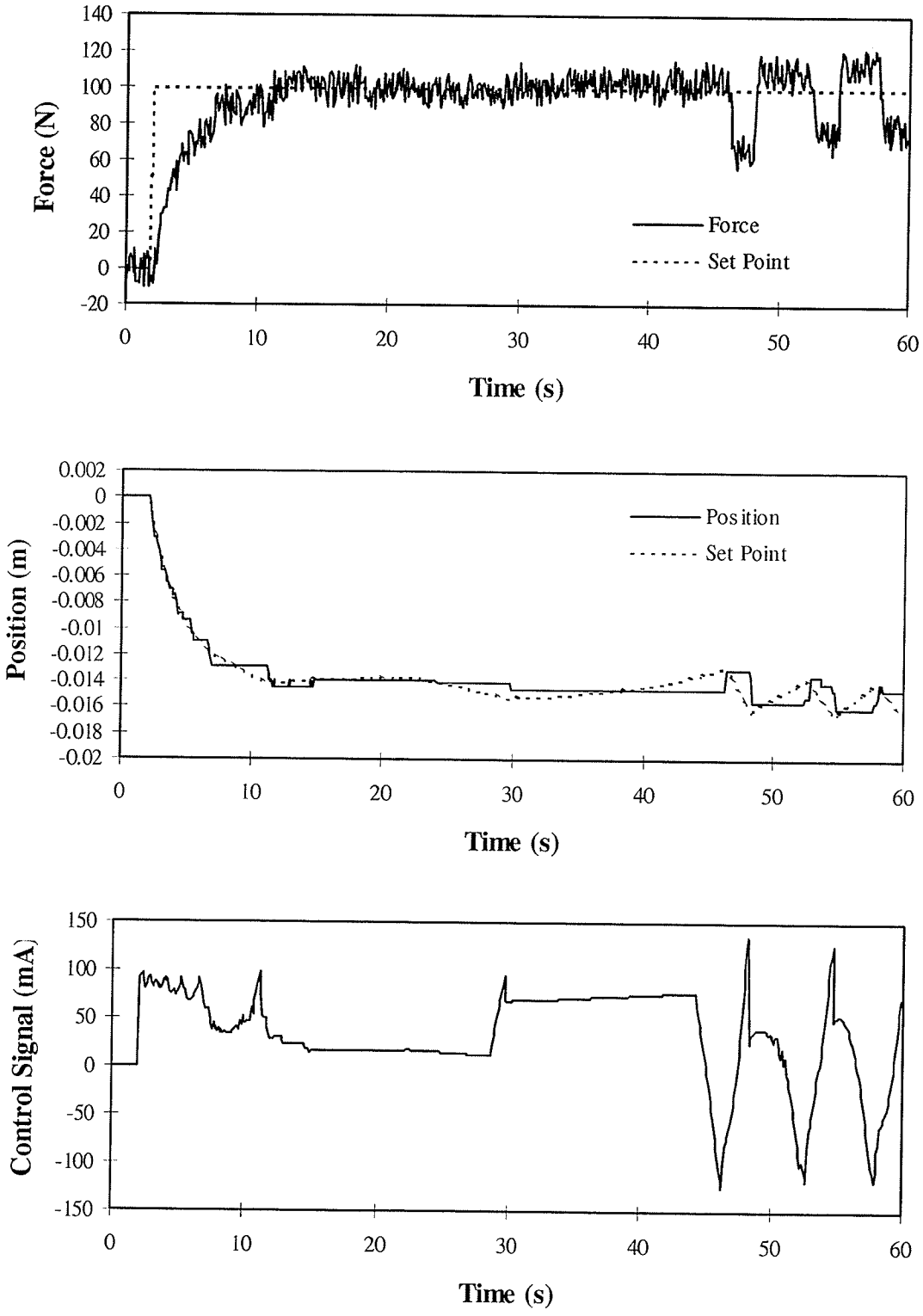


Figure 7-1: Prolonged static force control, m=500kg, ξ_{net}=5.0, k=50N/m.

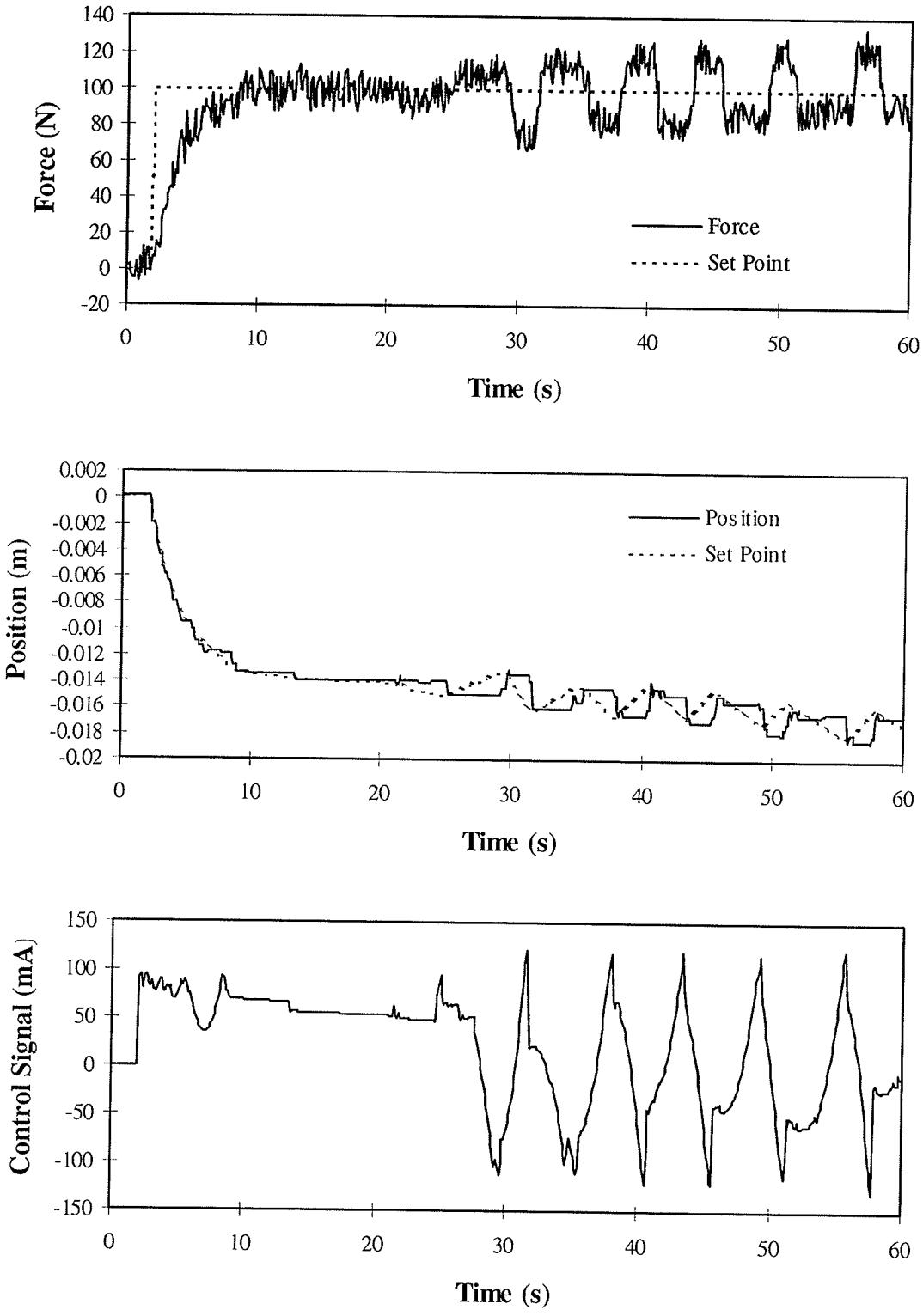


Figure 7-2: Prolonged static force control, m=500kg, ξ_{net}=5.0, k=500N/m.

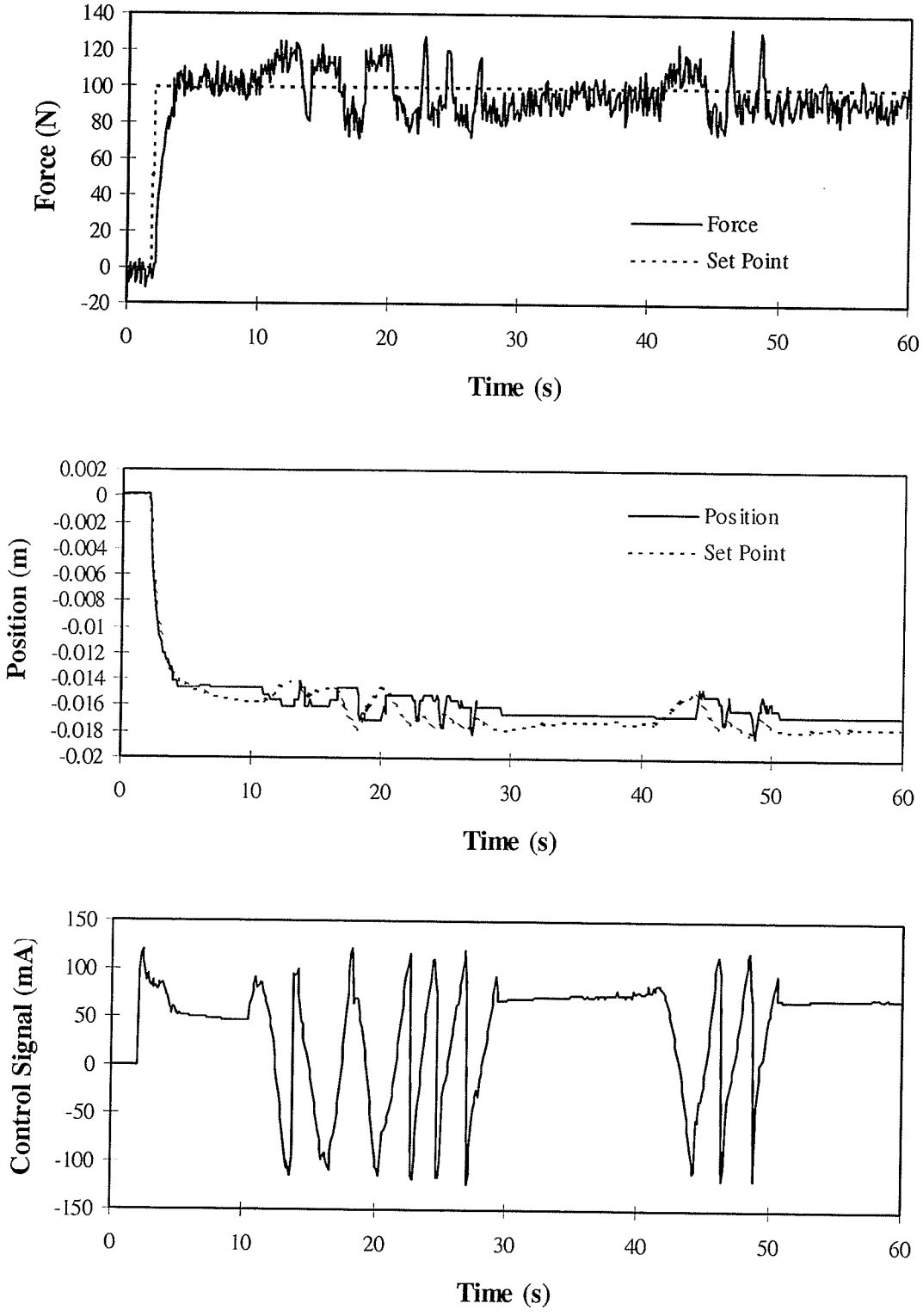


Figure 7-3: Prolonged static force control, m=500kg, ξ_{net}=2.0, k=5000N/m.

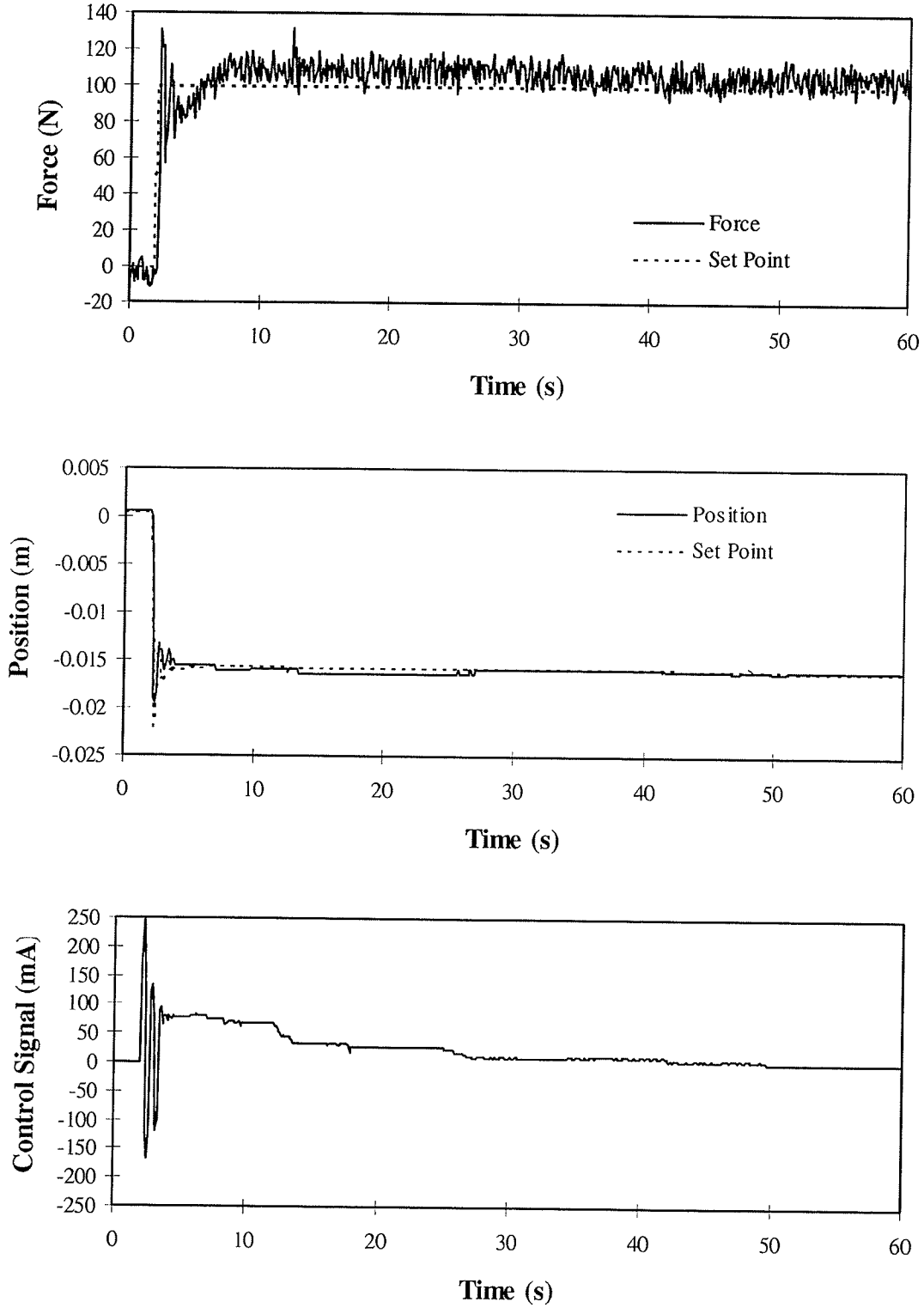


Figure 7-4: Prolonged static force control, m=500kg, ξ_{net}=0.3, k=50kN/m.

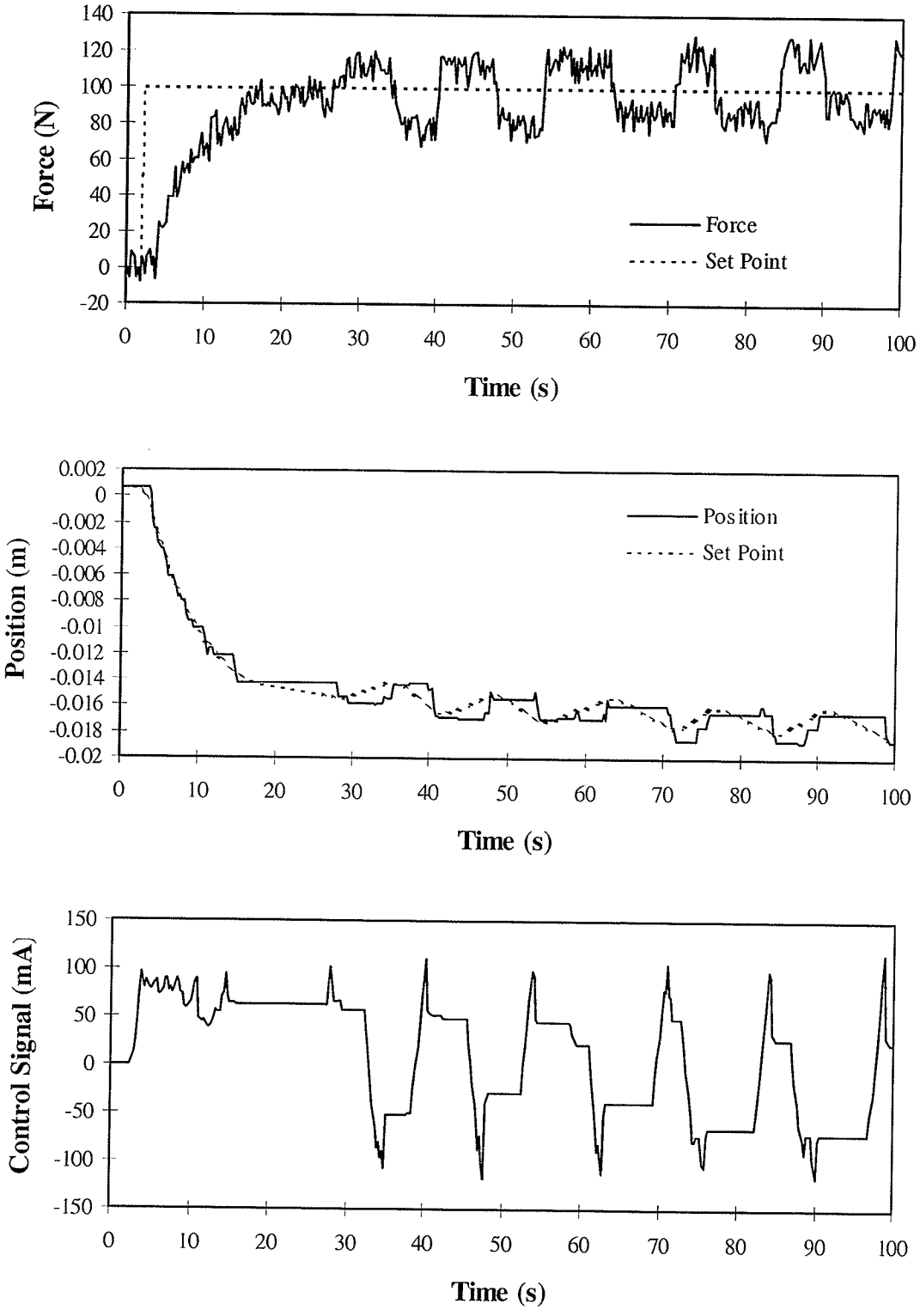


Figure 7-5: Prolonged static force control, m=50000kg, ξ_{net}=1.2, k=50N/m.

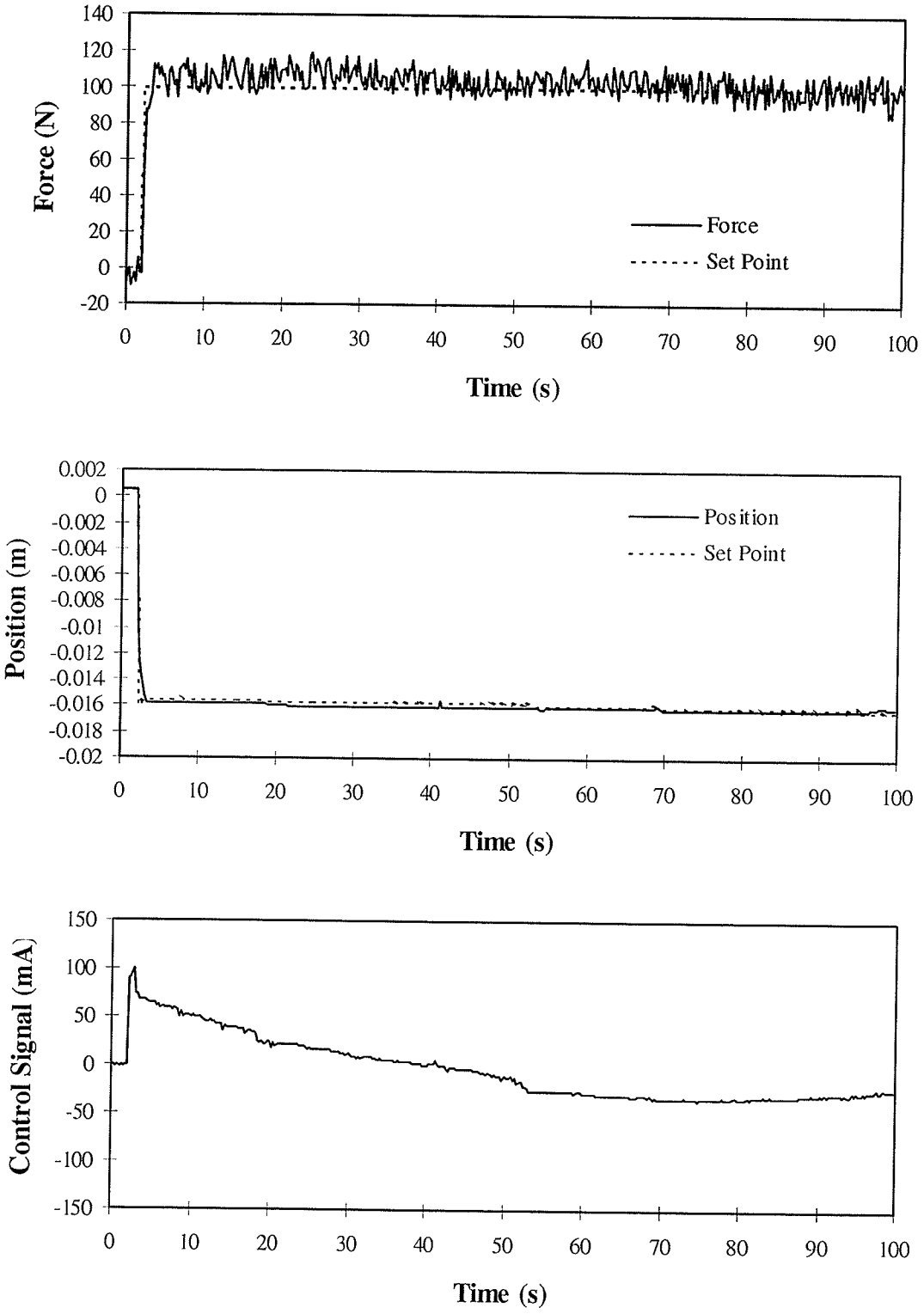


Figure 7-6: Prolonged static force control, m=50kg, ξ_{net}=1.2, k=50kN/m.

7.3 Consequences

The preceding analysis and experiments show that finite positioning accuracy results in limit cycling when the current formulation of PBIC is implemented. The only way to assure that these limit cycles will not occur is to set the target stiffness to be greater than double the effective environmental stiffness. This constraint has clear implications for static force control applications of PBIC. For stable operation, the required stiffness will also cause a large steady-state error, since the model stiffness will take the larger share of the virtually-imposed force set-point (see Figure 6-10). However, some form of set-point compensation can be used to regain accuracy. For example, multiplying the force set-point by $k_{eff}^{-1}(k+k_{eff})k^{-1}$ will completely compensate for load sharing. Even if the environmental stiffness is unknown, the force set-point can be safely multiplied by a factor of 2, so long as the environment is more compliant than the model.

The findings of this section lend further support to the newly-derived relation for force controller equivalence of Section 6.2. Figure 7-5 & Figure 7-6 especially show that the target mass does not affect the stability of the system in the manner predicted by the relation for torque based impedance control. All the experiments just shown confirm that instability is not immediately apparent, but becomes so in an amount of time according to the specified system dynamic. Recalling Figure 6-4, lower target stiffnesses will result in larger ultimate controller gains, and therefore instability, just as seen here.

7.4 Summary

A simple analysis has shown that any finite positioning accuracy will produce sustained oscillations in a position based impedance controlled system if the target stiffness is less than half the effective environmental stiffness. Limit cycles may yet occur unless the the target stiffness exceeds twice the effective environmental stiffness. Experiments confirmed these predicitions. The experiments also strengthened the evidence for the newly-derived relationship between position based impedanace control and explicit force control.

8. Conclusions

This work has addressed the problem of enabling an existing hydraulic manipulator to interact with a variety of environments. A review of the literature showed that a position based formulation of impedance control (PBIC) appears to be best suited to this application because of the difficulty in implementing torque control on hydraulic actuators.

Analysis of the control strategy showed that equivalent explicit force controllers exist for both torque and position based impedance controllers in a small, linearized region about the point of operation of the respective systems. This had already been shown for torque based formulations of impedance control, but a more general relation was derived here. The primary finding of this analysis, however, was that the form of the PBIC-equivalent explicit force controller was fundamentally different from the one already known for torque based impedance control (TBIC). The equivalent force control gains for PBIC were shown to vary inversely with target stiffness, while those for TBIC varied inversely with target mass. Furthermore, while the TBIC-equivalent controller was known to be a simple proportional controller, the PBIC-equivalent controller retained the form of its underlying position controller, modified by a second-order filter. These theoretical results were supported by experimental results contrasting the action of an impedance controller and its hypothetically equivalent explicit force controller.

To meet the demanding position-tracking requirements of this application, a novel position controller was developed. Based on a patented nonlinear PI controller by Morse et al. (1988), the new controller incorporated a nonlinear velocity error based signal that was shown to be able to reliably detect the onset of Coulomb friction-induced error and compensate for it. The controller developed was shown experimentally to be capable of accurate tracking of trajectories whose amplitudes exceeded 7 encoder resolution widths, and of regulation to within 2 encoder resolution widths of error. This level of accuracy was observed over 2 orders of magnitude of step, ramp, and sine wave trajectories in free space.

The position controller also proved to be robust in the presence of external forces applied at the end-effector, as seen in its performance as a component of the position based impedance control loop. Because of this continued level of performance, the impedance controller was shown experimentally to be capable of masking the actual, highly nonlinear robot dynamics and replacing them with those of the target impedance. The impedance controller was shown to be capable of static and dynamic force control, impact force reduction, and multi-dimensional coordinated position control and force regulation. These abilities were proven over nearly an order of magnitude of environmental stiffness.

Finally, an important condition on the long-term stability of the position based impedance controller with a finite positioning accuracy was shown. It was shown that limit cycles would result unless the target stiffness exceeded one half of the environmental stiffness, and could result unless the environmental stiffness was exceeded by a factor of at least two. These analytical predictions were experimentally confirmed. Furthermore, experiments showed that the only impedance parameter controlling the occurrence of limit cycles was the stiffness; increasing mass or damping would only delay the onset of the limit cycles, never prevent them. This finding further supported the newly derived relation for the PBIC-equivalent explicit force controller, since the TBIC-equivalent expression predicts increased stability with increased target mass. A simple method of retaining controller performance in force control tasks in the light of the restrictions on target stiffness was proposed and found to work well in experiment.

The above conclusions show that the theoretical and experimental work in this thesis have demonstrated the promise and limitations of position based impedance control for implementation on an existing industrial hydraulic manipulator.

9. Recommendations for Future Work

9.1 Theory

9.1.1 Optimal Form for Target Impedance

This work has focused on applications of second-order linear target impedances. This focus was made to facilitate comparison with the previous work of other researchers, the vast majority of whom used this model. However, there is no reason that another set of dynamic relations could not be used for the target impedance. Fore example, first order linear impedances have been used previously (Hogan, 1988; Johansson and Spong 1994; Cohen and Flash, 1991).

However, greater potential gains are to be made by examining the possibility of a general target impedance. As shown in Section 3.1.5, the target impedance can alternatively be thought of as an admittance parallel to the environmental admittance. Therefore, the apparent environmental admittance as seen by the end-effector is subject to modification based on the selection of the target impedance. The advantages and disadvantages of such modification bear examination.

9.2 Position Control

9.2.1 Robust Overshoot Reduction

The method of overshoot reduction employed in this work, the integration of braking accelerations, is effective but not robust. The amount of damping provided by this technique is proportional to the magnitude and duration of set-point deceleration, and therefore so to is the degree of overshoot reduction. Figure 4-25 shows that, in order to have good tracking at larger velocities, the overshoot reduction is insufficient at low velocities. Perhaps a velocity-dependent gain, or another method altogether, might prove more robust.

9.2.2 Reverse Repetition Motions

The dynamic force control test has identified a problem with the position controller: it had difficulty with fine control of motions when the required actuator torque opposed the direction of motion. The correct direction of motion was still obtained, but the trajectory exhibited overshoot, especially for small commanded displacements.

In order to extend the excellent trajectory-tracking abilities of the position controller to this scenario, some form of force feedback must be introduced. The equivalent joint torques caused by the forces on the end-effector can be obtained through the manipulator Jacobian. These joint torques could then be incorporated into the control signal, either by direct feedforward or through a more sophisticated technique.

9.3 Impedance Control

9.3.1 Noise Reduction

Presently, the high level of noise in the force sensor reading requires a target mass of at least 50kg for a smooth response. No efforts to reduce this level of noise were made because levels may be unavoidably this high in an actual application. However, if some industrially feasible hardware modification or filtering method were implemented, it would allow the controller to be tested using both faster target dynamics and lower target stiffnesses. It would be interesting to see how small of a transient response could be obtained in the interrupted circular trajectory test of Section 5.5. However, even with a successful reduction in noise, the fundamental constraints on the selection of stiffness would remain.

9.3.2 Circumventing Restrictions on Target Stiffness

The most serious problem with the position based impedance control method that has been uncovered in this work is the constraint on the selection of the target impedance for long-term stability. It should be noted that the effects of ignoring this restriction are only seen in prolonged experiments, unless fast target dynamics are specified. Limit cycles only appear if the force set-point and nominal position trajectory are held stationary for a

long time (compared with the characteristic time of the target impedance). Still, in some applications, the whole point of implementing impedance control is to increase the apparent compliance of the robot. If prolonged static environmental contact is required in such a situation, the necessary constraint of the target stiffness being greater than double the effective environmental stiffness will defeat the purpose of implementing the controller.

The force control experiments in Sections 5.3 & 5.4 showed that a set-point compensation method could be used to circumvent the effects of the stiffness restriction. It is worth investigating the possibility of using an analogous technique on the position set-point for implementation on problematic applications such as the one described above. For example, once environmental contact is confirmed by a non-zero force sensor reading, the incremental changes in the nominal trajectory along the constrained degree-of-freedom could be decreased by a factor calculated to give the correct steady-state displacement. For example, let k_- be the desired target stiffness which is less than the environmental stiffness, and k_+ be a stiffness greater than double the environmental stiffness assuring stability. Then defining Δx_d^* as the incremental change to the compensated nominal trajectory, using the following relation will mask the fact that k_+ is being used as the target impedance, making it appear as though k_- were used.

$$\Delta x_d^* = \frac{k_-}{k_+}\frac{k_+ + k_e}{k_- + k_e}\Delta x_d$$

This technique would have the same problems with estimation of environmental stiffness as the force set-point compensation method, but even the roughest estimation would provide an improvement over the uncompensated performance.

9.3.3 Alternate Formulations of PBIC

As shown in Section 3.1.5, the formulation of position based impedance control studied here is not the only possible formulation. The alternate formulation already given as an example could cause problems with the position controller developed herein because it incorporates its own position feedback, but it may have other advantageous properties.

For example, it is not known whether or not the constraint on target stiffness selection also applies to the alternate formulation.

Pelletier and Doyon (1994) suggested a number of possible formulations of impedance control for use with position based controllers. In fact, they concluded that an integrated velocity based formulation of PBIC had the best properties. It would be well worth investigating how their results apply to a hydraulic robot.

9.3.4 Adaptive Impedance Control

A vast and almost untapped area of research involves adaptively changing the target impedance parameters, or even the form of the target impedance, on-line to obtain optimal manipulator performance. All of the work presented here and most of the existing literature deal with the problem of making the manipulator impedance conform to a static target impedance. However, the models inspiring the whole field of robotics, organic limbs, are not controlled in this way. The wrist is made compliant to throw a baseball, and stiff to swing a bat. The impedance of the leg is varied greatly over one gait cycle (Hogan, 1985b). If robotic manipulators are to behave as naturally as organic ones, then the problem of controlling the robot to match a dynamic target impedance must be addressed.

As a preliminary investigation, a comparison of different simple adaptive techniques could be made. For example, the task of limiting the forces incurred in the trial of Section 5.5 to below an absolute threshold could be attempted by:

a) adaptively modulating the target stiffness,

b) adaptively modulating an offset to the position set-point, or

c) implementing a target impedance with a nonlinear "limit switch"-like spring.

The advantages and disadvantages of the different techniques could be contrasted, giving direction to the next step in the investigation.

References

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

2. An, C.H., and J.M. Hollerbach, "Kinematic Stability Issues in Force Control of Manipulators," *Proceedings of the IEEE Conference on Robotics and Automation*, 1987, pp. 897-903.

3. Beer, F.P. and E.R. Johnston Jr., *Mechanics of Materials*, McGraw-Hill, New York, 1981, p. 598.

4. Chen, Y.N., C.B. Lee and C.H. Tseng, "A Variable Structure Controller Design for an Electro-Hydraulic Force Control Servo System," *Journal of the Chinese Society of Mechanical Engineers*, Vol. 11, No. 6, 1990, pp. 520-526.

5. Cohen, M., and T. Flash, "Learning Impedance Parameters for Robot Control Using an Associative Search Network," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, June 1991, pp. 382-389.

6. Conrad, F. and C.J.D. Jensen, "Design of Hydraulic Force Control Systems with State Estimate Feedback," *Proceedings of the IFAC 10th Triennial World Congress*, Munich, FRG, 1987, pp. 307-312.

7. Dawson, D.M., F.L. Lewis and J.F. Dorsey, "Robust Force Control of a Robot Manipulator," *The International Journal of Robotics Research*, Vol. 11, No. 4, August 1992, pp. 312-319.

8. Denavit, J., and Hartenburg, R.S., "A Kinematic Notation for Lower-Pair Matrices Based on Matrices," Journal of Applied Mechanics, Vol. 77, 1955, pp. 215-221.

9. D'Souza, A.F., *Design of Control Systems*, Prentice-Hall, New Jersey, 1988, p. 240.

10. Eppinger, S.D., and W.P. Seering, "On Dynamic Models of Robot Force Control," *Proceedings of the IEEE Conference on Robotics and Automation*, 1986, pp. 29-34.

11. Field, G., and Y. Stepanenko, "Model Reference Impedance Control of Robotic Manipulators," *Proceedings of the IEEE Pacific Rim Conference*, 1993, pp. 614-617.

12. Fisher, W.D. and M.S. Mujtaba, "Hybrid Position/Force Control: A Correct Formulation," *The International Journal of Robotics Research*, Vol. 11, No. 4, August 1992, pp. 299-311.

13. Fu, K.S., R.C. Gonzales and C.S.G. Lee, *Robotics: control, sensing, vision, and intelligence*, McGraw-Hill, New York, 1987, p. 38.

14. Goldenberg, A.A., "Implementation of Force and Impedance Control in Robot Manipulators," *Proceedings of the IEEE Conference on Robotics and Automation*, 1988, pp. 1626-1632.

15. Hanafusa, H., H. Asada and T. Mikoshi, "Design of Electrohydraulic Servo Systems for Articulated Robot Arm Control," *IFAC Pneumatic and Hydraulic Components and Instruments in Automatic Control*, 1980, pp. 223-228.

16. Hogan, N., "Impedance Control: An Approach to Manipulation Part I - Theory," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 107, March 1985, pp. 1-7.

17. Hogan, N., "Impedance Control: An Approach to Manipulation Part II - Implementation," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 107, March 1985, pp. 8-16.

18. Hogan, N., "Impedance Control: An Approach to Manipulation Part III - Applications," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 107, March 1985, pp. 17-24.

19. Hogan, N., "Stable Execution of Contact Tasks using Impedance Control," *Proceedings of the IEEE Conference on Robotics and Automation*, 1987, pp. 1047-1054.

20. Hogan, N., "On the Stability of Manipulators Performing Contact Tasks," *IEEE Journal of Robotics and Automation*, Vol. 4, No. 6, December 1988, pp. 677-686.

21. Johansson, R. and M.W. Spong, "Quadratic Optimization of Impedance Control," *Proceedings of the IEEE Conference on Robotics and Automation*, 1994, pp. 616-621.

22. Lasky, T.A., and T.C. Hsia, "On Force-Tracking Impedance Control of Robot Manipulators," *Proceedings of the IEEE Conference on Robotics and Automation*, 1991, pp. 274-280.

23. Liu, G.J., and A.A. Goldenberg, "Robust Hybrid Impedance Control of Robot Manipulators," *Proceedings of the IEEE Conference on Robotics and Automation*, 1991, pp. 287-292.

24. Lu, W.S., and Q.H. Meng, "Impedance Control with Adaptation for Robotics Manipulations," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, June 1991, pp. 408-415.

25. Luh, J.Y.S., M.W. Walker and R.C. Paul, "Resolved-Acceleration Control of Mechanical Manipulators," *IEEE Transactions on Automatic Control*, Vol. 25, No. 3, June 1980, pp. 468-474.

26. McCormick, W., and H.M. Stewards, "An Investigation of Impedance Control for Robot Manipulators," *International Journal of Robotics Research*, Vol. 12, No. 5, October 1993, pp. 473-489.

27. Morse, R.A., C.P. Day, and K. Troy, "Positional Control Method and System Utilizing Same," United States Patent Number 4727303, Feb. 23, 1988.

28. Pannala, A.S., P. Dransfield, M. Palaniswami and J.H. Anderson, "Controller Design for a Multichannel Electrohydraulic System," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 111, June 1989, pp. 299-306.

29. Pelletier, M., and M. Doyon, "On the Implementation and Performance of Impedance Control on Position Controlled Robots," *Proceedings of the IEEE Conference on Robotics and Automation*, 1994, pp. 1228-1233.

30. Raibert, M.H., and J.J. Craig, "Hybrid Position/Force Control of Manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 102, June 1981, pp. 126-133.

31. Salisbury, J.K., "Active Stiffness Control of a Manipulator in Cartesian Coordinates," *Proceedings of the 19th IEEE Conference on Decision and Control*, 1980, pp. 95-100.

32. Tsai, C.K. and D.E. Orin, "Modified Hybrid Control for an Electro-Hydraulic Robot Leg," *Proceedings of the ASME International Computers in Engineering Conference and Exhibition*, July 1986, Vol. 1, pp. 95-100.

33. Volpe, R., and P. Khosla, "A Theoretical and Experimental Investigation of Explicit Force Control Strategies for Manipulators," *IEEE Transactions on Automatic Control*, Vol. 38, No. 11, November 1993, pp. 1634-1650.

34. Volpe, R., and P. Khosla, "A Theoretical and Experimental Investigation of Impact Control for Manipulators," *International Journal of Robotics Research*, Vol. 12, No. 4, August 1993, pp. 351-365.

35. Zhang, H. and R.P. Paul, "Hybrid Control of Robot Manipulators," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1985, pp. 80-85.

Appendix A: Inverse Kinematics for the Unimate MKII-2000

The inverse kinematics for the Unimate were derived using the descriptive conventions established by Denavit and Hartenburg. The manipulator parameters are (Fu et al., 1987):

Table 1: Denavit-Hartenburg parameters for Unimate MKII-2000.

Link	Variable	θ_i	α_i	a_i	d_i
1 (Right/Left)	θ_1	-90°	-90°	0	d_1
2 (Down/Up)	θ_2	-90°	90°	a_2	0
3 (Out/In)	d_3	-90°	0°	0	d_3

Using these parameters, the following transformation matrices are obtained:

$$ {}^{0}\mathbf{A}_{1} = \begin{bmatrix} \cos\theta_1 & 0 & -\sin\theta_1 & 0 \\ \sin\theta_1 & 0 & \cos\theta_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{1}\mathbf{A}_{2} = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 & a_2\cos\theta_2 \\ \sin\theta_2 & 0 & -\cos\theta_2 & a_2\sin\theta_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{2}\mathbf{A}_{3} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Multiplying these matrices gives the following set of relations for the end point position:

$$x = \cos\theta_1\{d_3 \sin\theta_2 + a_2 \cos\theta_2\}$$
$$y = \sin\theta_1\{d_3 \sin\theta_2 + a_2 \cos\theta_2\}$$
$$z = d_1 + d_3 \cos\theta_2 + a_2 \sin\theta_2$$

Solving these equations for the joint variables gives the inverse kinematic solution:

$$\theta_1 = \tan^{-1}\left(\frac{y}{x}\right)$$
$$\theta_2 = \tan^{-1}\left(\frac{d_3\sqrt{x^2+y^2} - a_2(z-d_1)}{a_2\sqrt{x^2+y^2} - d_3(z-d_1)}\right)$$
$$d_3 = \sqrt{x^2+y^2+(z-d_1)^2 - a_2^2}$$

Note that this solution is unambiguous (one-to-one and onto) in its mapping from Cartesian space into joint space.

Appendix B: Velocity Measurement by Linear Regression of Digital Encoder Readings

Problem

In the absence of tachometers, the joint velocities required by the controller must be obtained using the position readings from the encoders and the times at which those readings were obtained. A simple linear regression can be used to estimate the slope of the position history. However, this simple technique can be complicated by the finite encoder resolution.

At low joint velocities and/or high sampling rates, the joint movement between subsequent readings may be less than the encoder can measure, resulting in identical position measurements. In the limiting case of a 2-point regression, this phenomenon will result in the calculation of a false, zero velocity for the second such point. When the encoder resolution is eventually surpassed, a spike in velocity will occur. At the other extreme, using many points in the regression will eliminate this impulse behaviour, but can smooth the profile excessively, obscuring detail. Furthermore, because only the points on one side of the instant can be used in the regression, a time lag is introduced.

It is therefore necessary to find a functional trade-off between the two extremes. The number of points used for this optimal regression will depend on the sampling rate, the operating range of velocities, and the requirements of the system.

Linear Regression

The best linear fit to a set of data points is the vector $\{v\ x_o\}^T$ that minimizes the sum of the squared errors in the solution of the following equation:

$$\begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} = \begin{bmatrix} t_1 & 1 \\ t_2 & 1 \\ \vdots & \vdots \\ t_n & 1 \end{bmatrix} \begin{Bmatrix} v \\ x_o \end{Bmatrix}$$

where n is the number of points used in the regression. This equation can be written more compactly as:

$$\mathbf{x} = \mathbf{T}\mathbf{v}$$

The solution for the linear parameters v and x_o that results in the least squared error is given as follows:

$$\hat{\mathbf{v}} = \left(\mathbf{T}^T\mathbf{T}\right)^{-1}\mathbf{T}^T\mathbf{x}$$

Experimental Results

A position trajectory was generated using a quarter cosine wave of amplitude 10 degrees. An encoder resolution of 0.28 degrees and sampling rate of 200_{Hz} were assumed, and used to digitize the trajectory. The actual and digitized trajectories are shown in Figure 9-7. The position was assumed to be zero for all negative time, for purposes of the regression.

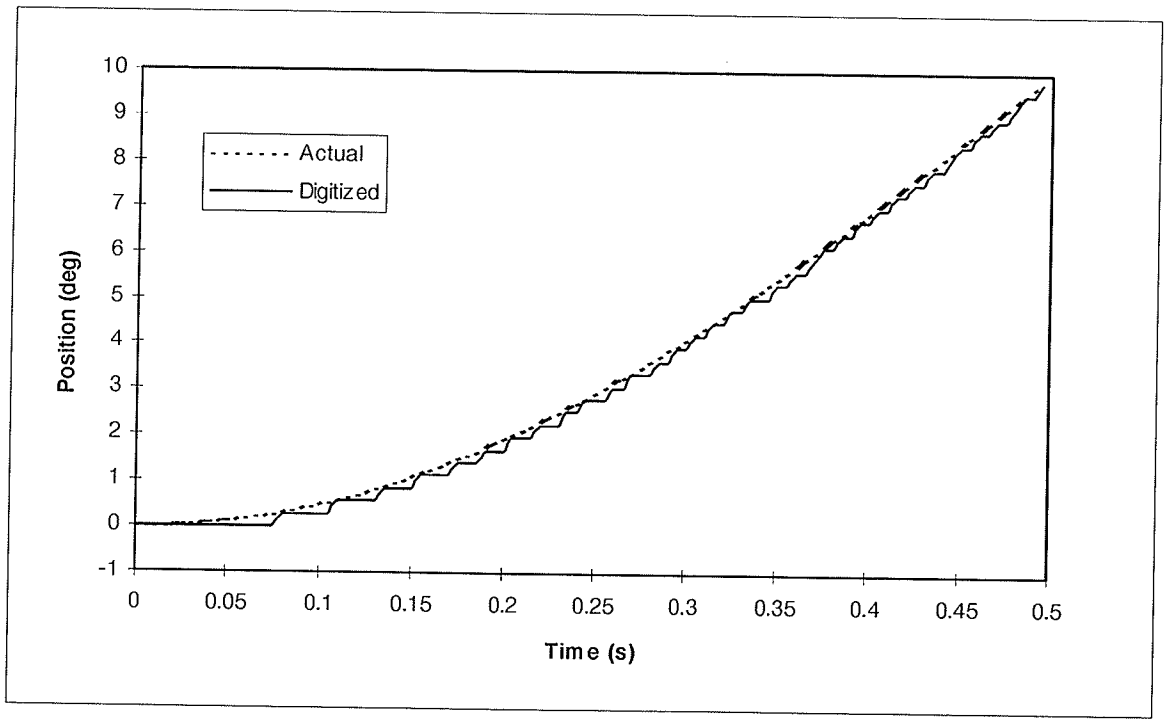


Figure 9-7: Test position profile.

The results of using different number of points in the regression are shown in Figure 9-8. The impulse behaviour previously described is clearly visible when using a 2-point regression. Much oscillation is present even with a 10 point regression. The derived velocity profile is reasonably smooth using a 20 point regression, but a more noticeable lag is introduced. This lag increases as more points are used in the regression. Eventually, at 100 points, excessive smoothing has greatly distorted the character of the velocity profile.

Based on this analysis, a 20 point regression was used to obtain the velocities required by this controller. The lack of significant artificial oscillations was important for stability when using velocity error based control signals. Equally important, however, was a short lag time, in order to ensure a quick response in stiction detection. A short lag time was also critical so that no significant phase lag develops between position and velocity signals.

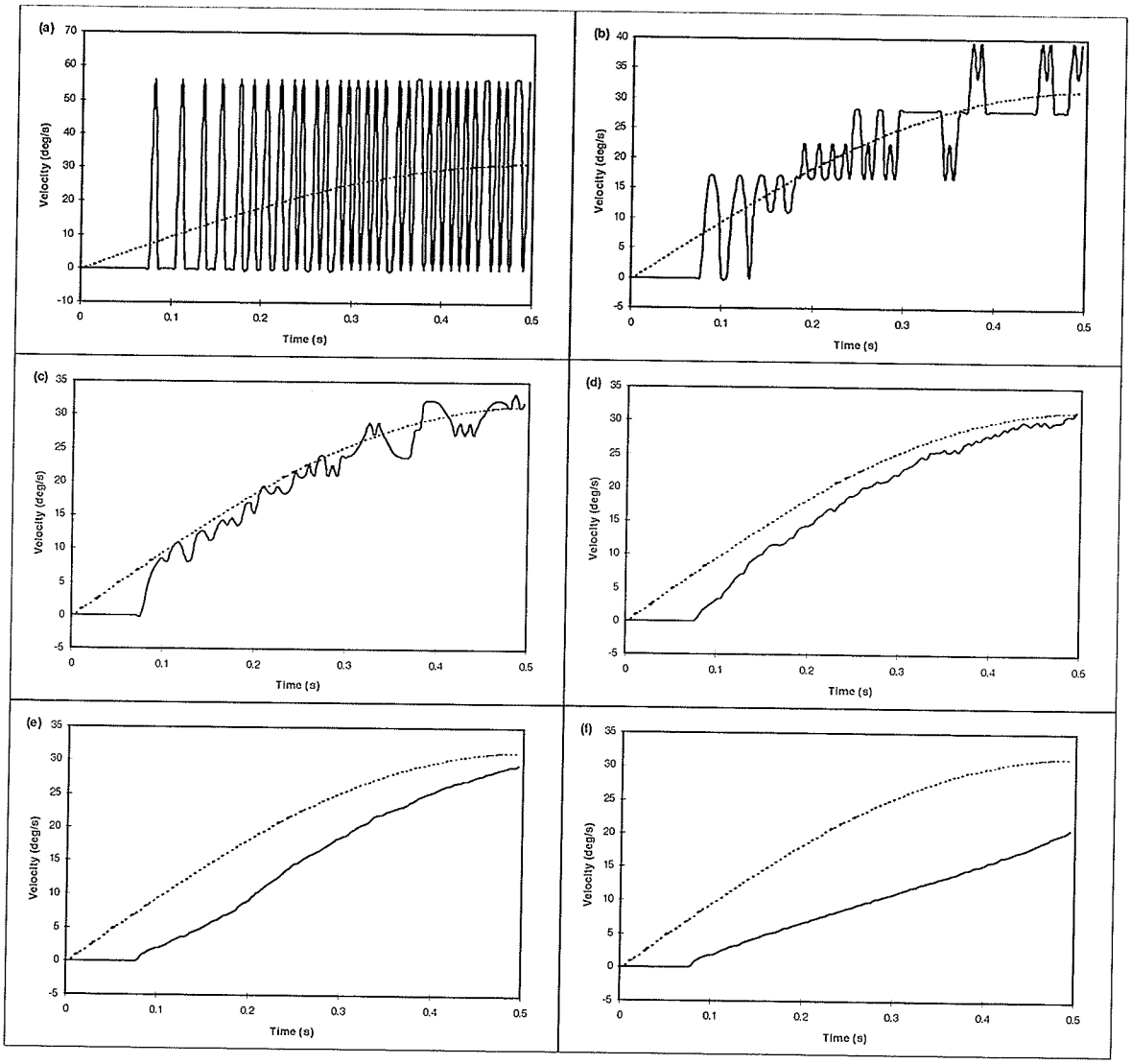


Figure 9-8: **Velocity profiles obtained using 2,5,10,20,40, and 100 point regressions (a-f, respectively).**

Appendix C: Position Controller Tuning and Gains

The total position control action is described by the following equations:

$$u(t) = K_P e + K_I I(t)$$

$$e(t) = q_s(t) - q(t)$$

$$I(t) = \begin{cases} \dfrac{-U_{lower} - K_P e(t)}{K_I}, \; flag = 0 & \dot{e}_{stiction} < -\dot{e}_{min} \;\&\; U > -U_{lower} \;\&\; flag = 1 \\ \left(I(t-\Delta t) + e(t)\Delta t + K_a \ddot{q}^{braking}\Delta t\right)\dfrac{\alpha(\dot{q})}{\alpha(\dot{q}) + \dot{e}^2(t)}, \; flag = 1 & \left|\dot{e}_{stiction}\right| \le \dot{e}_{min} \;\&\; flag = 0 \\ \left(I(t-\Delta t) + e(t)\Delta t + K_a \ddot{q}^{braking}\Delta t\right)\dfrac{\alpha(\dot{q})}{\alpha(\dot{q}) + \dot{e}^2(t)}, \; flag = 0 & \left|\dot{e}_{stiction}\right| \ge \dot{e}_{min} \;\&\; flag = 0 \\ \dfrac{U_{upper} - K_P e(t)}{K_I}, \; flag = 0 & \dot{e}_{stiction} > \dot{e}_{min} \;\&\; U < U_{upper} \;\&\; flag = 1 \end{cases}$$

$$\dot{e}(t) = \dot{q}_s(t) - \dot{q}(t)$$

$$\alpha(\dot{q}) = \alpha_{\max}\left(0.001 + \frac{|\dot{q}|}{\dot{q}_{\max}}\right)$$

$$\ddot{q}^{braking} = \begin{cases} \ddot{q} & \text{if } \ddot{q}\dot{q} < 0 \\ 0 & \text{if } \ddot{q}\dot{q} \ge 0 \end{cases}$$

$$\dot{e}_{stiction} = \dot{e}\frac{\dot{q}_s^2}{\dot{q}_s^2 + \beta\dot{q}^2}$$

The position controller gains were tuned empirically using the test patterns plotted in Section 4 to assess performance. The following procedure was used:

1. Establish maximum stable proportional gain using linear P-control.
2. Add integral control action, determining maximum integral gain.
3. Convert to modified Morse controller, determine optimum α and K_I for fastest ramp test (the parameter β was always set at 50).
4. Set α_{max} to this value of α.

5. Determine threshold where controller action overcomes stiction and deadband for each direction of motion. Use thresholds in VETIA method.

6. Add braking acceleration to the integral. Determine optimum K_a for overshoot reduction using ramp and sine wave test patterns.

This procedure was applied to each link of the Unimate in isolation. The results are summarized below:

Table 2: Position controller gains.

Control Gain	Link 1 (left/right)	Link 2 (up/down)	Link 3 (in/out)
K_P	92 mA·deg^{-1}	215 mA·deg^{-1}	17600 mA·m^{-1}
K_I	613 mA·deg^{-1}s^{-1}	1151 mA·deg^{-1}s^{-1}	65900 mA·m^{-1}s^{-1}
α_{max}	5000 deg^{2}s^{-2}	8000 deg^{2}s^{-2}	20 m^{2}s^{-2}
$\dot{q}_{max}$	10 deg/s	10 deg/s	0.2 m/s
$\dot{e}_{min}$	0.15 deg/s	0.15 deg/s	0.0015 m/s
U_{lower}	88 mA	88 mA	141 mA
U_{upper}	88 mA	114 mA	176 mA
K_a	26.4 s^2	150 s^2	3520 s^2
β	50	50	50

Appendix D: Plate Stiffness

The equivalent Cartesian stiffness of the test plate can be found using standard beam deflection theory (Beer and Johnson, 1981).

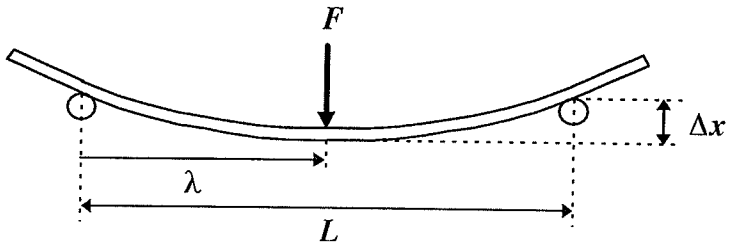


Figure 9-1: Test plate under load.

The deflection of the plate at the point λ of application of the load F is:

$$\Delta x = \frac{F\lambda^2(L-\lambda)^2}{3EIL}$$

where E is Young's modulus for the material and I is the moment of inertia of the plate's cross section about its axis of rotation. The apparent Cartesian stiffness of the plate is:

$$\begin{aligned} k_e &= \frac{F}{\Delta x} \\ &= \frac{3EIL}{\lambda^2(L-\lambda)^2} \\ &= \frac{48EI}{L^3} \end{aligned}$$

for $\lambda = 0.5L$. The plate was made of aluminum (E = 70GPa), was of rectangular cross-section (width 10.5in., thickness 1/8in.) with supports spaced by 27.25in. or 14in. These numbers give stiffnesses of 7.2kN/m and 53kN/m for the two spacings, respectively. Analysis of the above equation shows that these stiffness values are fairly insensitive to errors in positioning the load. Placing the load 5% to one side of center results in a

stiffness which is only 2% higher than that calculated for a centrally applied load, while a 10% error in load placement results in a 8% error in stiffness.

Appendix E: Sensor Stiffness

The sensor stiffness was calculated using a linear regression on data collected from the experimental apparatus. The manipulator was positioned so that it was exerting a downwards force on a stiff object, such as a chair. The magnitude of the force across the force sensor was then calculated, and the perpendicular surface-to-gripper distance was measured manually. The measurements varied by up to 0.5 inches between trials at a given force level due to compliance in the gripper, which allowed some variation in the angle at which the sensor assembly was held. This variation was not compensated for (except through statistical methods) because the phenomenon was inherent to the apparatus, and would therefore occur during all experiments. The results of this measurement procedure are shown below.

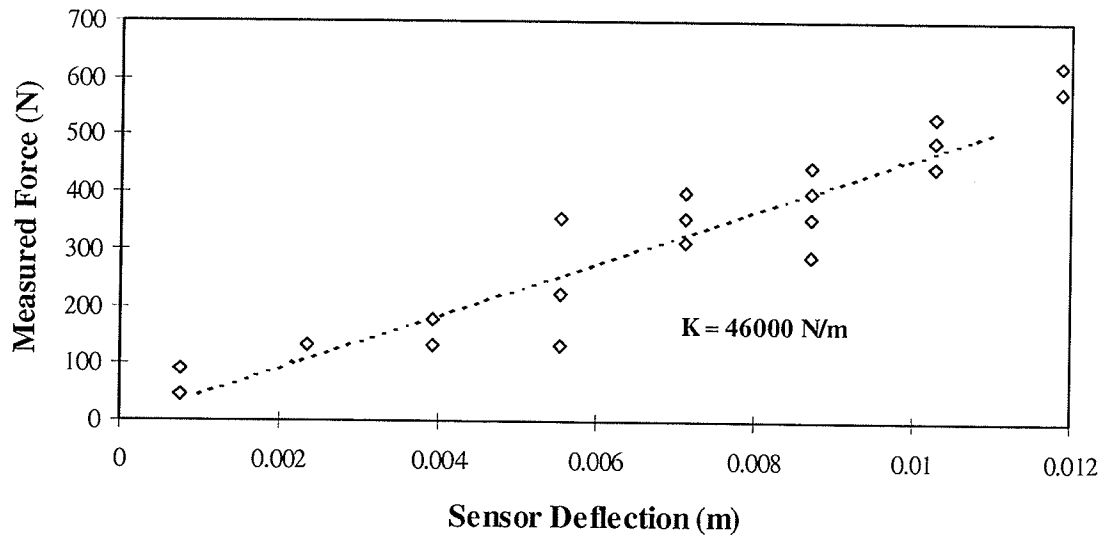


Figure 9-2: Linear regression on sensor deflection data.

Appendix F: Code

```
/* imptrial.c                                                          */
/* Model Reference Impedance Controller for the Unimate Robot          */
/* Written 04/95 to 10/95                                              */
/* Bradley Heinrichs                                                   */
/* Some routines by A. Lohse                                           */

#include <stdio.h>
#include <conio.h>
#include <process.h>
#include <dos.h>
#include <math.h>
#include <string.h>
#include <malloc.h>

#define TimerResolution  1193181.667
#define D_TO_A_BASE_ADDRESS 704
#define A_TO_D_BASE_ADDRESS 720
#define PIO12_BASE_ADDRESS  736
#define Tsample 0.005
#define DTSAVE 0.05                  /* in seconds */
#define DIMSAVE 1000                 /* number of data points to store */
#define N_reg 20                     /* Use N_reg = 20/(200*Tsample) */
#define MAX_REG 2
#define IND_t 0
#define IND_1 1
#define IND_2 2
#define IND_3 3
#define IND_1d 4
#define IND_2d 6
#define IND_3d 9
#define IND_u2 5
#define IND_force 7
#define IND_fd 8
#define IND_k 10
#define IND_MAX 12

FILE *zout,*zdout;
FILE *fout,*fdout;
FILE *xlout,*xl2out,*xl3out;

int d_to_a_channel,data;
int mask[6] = {0,31,7,3,7,3};
long t1,t2;
double r,time=0.0,oldtime=0.0;
int ndata=0;
float data_block[IND_MAX][DIMSAVE];

float *x,*xd,*xi,*qd,*qddot,*q,*qdot,*qi,*u,*fd;
float *HighLimit,*LowLimit;
float a2=0.13,a4=0.5,q4;             /* Fixed manipulator parameters */
float *fend;                         /* External force on end-effector */
float *Kp,*Ki,*Ka,*alpha;            /* Controller gains */
float k,k_i,c,m,eta,eta_net,wn;      /* Target impedance */
float kenv;
float deltat;                        /* Interval between samples (sec) */
float uboost,foffset=30.0;

float rtd=180.0/3.14159;

char savedir[60];

void store_data(int i);
void save_data(void);
FILE* openfile(char *filename, char *mode);
void open_files(char *dir);
void close_files(void);
void initializetimer(void);
void restoretimer(void);
long readtimer(void);
int get_a_to_d(int a_to_d_channel);
```

```
  void initialize_d_to_a(void);
  void initialize_pio12(void);
  void out_d_to_a(int i);
  void zero_voltage_and_current_outputs(void);
  void get_foffset(void);

  void path(float *q_des,float *qdot_des,float pos_min,float pos_max,float T[],
                float t_traj[],float t_acc,float time);
  void trajectory(float qi,float qf,float T,float t_acc,float t_rel,float *q_des,
                float *qdot_des);
  double regress_velocity1(double time, double x);
  void regress_velocity_vector(double time, float *x, float *v, int i);
  int morse_cont(float *state,float *state_d, float h, float *output);
  int equiv_f_cont(float *state,float *state_d, float h, float *output);
  int PID_cont(float *state,float *state_d, float h, float *output);
  float ROI_cont(float control_signal, float vel, float time);
  float anti_stick_boost(float v, float v_set);
  void set_qd(float *q_d, float time);
  void set_fend(float *f, float time);
  void qtox(float *q, float *x);
  void xtoq(float *x, float *q);
  void nrerror(char *error_text);
  void free_vector(float *v,long nl,long nh);
  float *vector(long nl,long nh);
  void set_parameters();

 main(int argc, char *argv[])
 {
   int i=0;

   set_parameters();
   zero_voltage_and_current_outputs();
   initialize_pio12();
   initialize_d_to_a();
   strcpy(savedir,argv[1]);

   get_foffset();

   initializetimer();
   write_information_to_monitor();

   while(!kbhit() && ndata<DIMSAVE && time < 15.0) {

         t2 = readtimer();
         r = (( (t2-t1) < 0) ? 4294967296.0 + (long) (t2-t1) : (long) (t2-t1));
         deltat= r/TimerResolution-time;
         time = r / TimerResolution;

         trigger_switch();
         display_joint_coordinates();
         display_force_gauge();
         joystick_manipulate();
         if(deltat<0.001) store_data(ndata++);
         if((time-oldtime)>DTSAVE) {
               store_data(ndata++);
               oldtime+=DTSAVE;
         }
         t2 = readtimer();
         r = (( (t2-t1) < 0) ? 4294967296.0 + (long) (t2-t1) : (long) (t2-t1));
         deltat= r/TimerResolution-time;
         r=(Tsample>deltat) ? (Tsample-deltat)*1000.0: 0.0;
         delay((unsigned)r);
         i++;
   }
   zero_voltage_and_current_outputs();
   save_data();
   restoretimer();
}

/*********************************************************************/
display_force_gauge()
{
        int value;
        float dz, xo;
```

```
        value = get_a_to_d(15);

        fend[1]=0.0;
        fend[2]=0.0;
        fend[3]=value*20.0/66.0*9.8/2.2;  // Newtons (I think)
        xo=0.0;
        fend[3]-=foffset;

        // Failsafe in case of high forces

        if(fend[3]>75.0*9.8/2.2) {
                zero_voltage_and_current_outputs();
                save_data();
                restoretimer();
                exit(1);
        }
}
/* ********************************************************************** */
display_joint_coordinates()
{
        int lsb,msb,angle,i;
        int j[6] ={0,1,3,2,5,4};
        static int flag=1;
        gotoxy(1,17);
        for (i=1;i<=5;i++)
        {
                outportb(PIO12_BASE_ADDRESS+2,j[i]);        /*choose input channel */
                lsb=inp(PIO12_BASE_ADDRESS);
                msb=inp(PIO12_BASE_ADDRESS+1);
                angle = lsb+256 *(msb & mask[j[i]]);

                if(i==1) {
                        q[1]=0.0281778*((float) angle) - 113.5848;
                }
                if(i==3) {
                        q[2]=-0.0284006*((float) angle) + 27.59;
                }
                if(i==2) {
                        q[3]=0.00105*((float) angle) - 0.0052;
                }
                if(i==5) {
                        q4=-0.113475*((float) angle) + 3.797;
                }
        }
        qtox(q,x);
        if(flag) {
                for(i=0;i<3;i++) qi[i]=q[i];
                qtox(q,xi);
                flag=0;
        }

}

print_binary(int byte)
        {
        int i;
        for (i=7;i>=4;i--)
                {
                if(byte & (int)pow(2.0,i) )putch('1');
          else putch('0');
                }
          putch(' ');
        for (i=3;i>=0;i--)
                {
                if(byte & (int)pow(2.0,i) )putch('1');
          else putch('0');
                }
          putch(' ');
          }

joystick_manipulate()
{
        int x,y,z,w,v,r;
        int i;
```

```
        w = 2*get_a_to_d(3);  /* small j/s twist */
        v = 2*get_a_to_d(4);  /* small j/s up/down */
        r = -2*get_a_to_d(5); /* small j/s left/right */

        x=0;            /* Link 1 */
        y=0;            /* Link 3 */
        z=0;            /* Link 2 */
        set_qd(qd,time);
        regress_velocity_vector(time,q,qdot,0);
        regress_velocity_vector(time,qd,qddot,1);
        morse_cont(q,qd,deltat,u);                      // Use morse or equiv_f, not both
//      equiv_f_cont(fend,fd,deltat,u);
        for(i=1;i<=3;i++) {
                if (u[i]>2048.0) u[i]=2048.0;   // Limit output
                if (u[i]<-2048.0) u[i]=-2048.0;
        }
        x=(int)u[1];
        y=(int)u[3];
        z=(int)u[2];

        data = x+2048;    d_to_a_channel = 2;     /* 3 => X-valve,rotate */
        out_d_to_a(data);
        data = y+2048;    d_to_a_channel = 5;    /* 4 =>  Y-valve,extend */
        out_d_to_a(data);
        data = z+2048;    d_to_a_channel = 1;    /* 1 =>  Z-valve,up/down */
        out_d_to_a(data);
        data = r+2048;    d_to_a_channel = 3;    /*5 =>  W-valve ,roll*/
        out_d_to_a(data);
        data = v+2048;    d_to_a_channel = 4;    /*2 =>  U-valve ,pitch*/
        out_d_to_a(data);
        }
/*****************************************************************************/
float braking_acceleration(int link)
{
        static float vo[4]={0.0,0.0,0.0,0.0};
        float acc, maxacc[4]={0.0,15.0,100.0,0.05};;

        acc=(qddot[link]-vo[link])/deltat;   // Assume called every iteration
        vo[link]=qddot[link];
        if(acc>0.0 && qddot[link]<0.0) {
                if(acc>maxacc[link]) acc=maxacc[link];
        } else  if(acc<0.0 && qddot[link]>0.0) {
                if(acc<-maxacc[link]) acc=-maxacc[link];
        } else acc=0.0;
        return acc;
}
/*****************************************************************************/
int morse_cont(float *state,float *state_d, float h, float *output)
{
        float minerr[4]={0.0,0.03,0.03,0.00105};
        static float intg[4]={0.0,0.0,0.0,0.0};
        static flag[4]={1,1,1,1};
        float boostthresh[4]={0.0,0.15,0.15,0.0015};
        float deadcomp[4]={0.0,450.0,450.0,650.0};
        float deaddown[4]={0.0,500.0,500.0,800.0};
        float deadup[4]={0.0,500.0,650.0,1000.0};
        float qddotmax[4]={0.0,10.0,10.0,0.2};
        float err,vel,nonlin,a,ub;
        int i;

        for(i=1;i<=3;i++) {
                err = state_d[i]-state[i];
                vel=qddot[i]-qdot[i];    // Note: vel is in deg/sec
                if(fabs(err)<2.0*minerr[i] && fabs(qddot[i])<minerr[i]) {
                        err=0.0;        // Don't adjust very small errors
                }                       // (leads to random walk in intg)
                a=alpha[i]*(0.001+fabs(qddot[i]/qddotmax[i]));
                nonlin=a/(a+vel*vel);
                intg[i] += Ki[i]/4.0*err*h;//+Ka[i]*braking_acceleration(i)*h;
                intg[i] *= nonlin;
                output[i] = Kp[i]*err + intg[i];

                // Enable stiction compensation

                ub=anti_stick_boost(qdot[i],qddot[i]);
```

```
                if(ub>boostthresh[i]) {
                        if(flag[i] && (output[i]<deadcomp[i])) {
                                intg[i]+=deaddown[i]-output[i];
                                flag[i]=0;
                        }
                } else if(ub<-boostthresh[i]) {
                        if(flag[i] && (output[i]>-deadcomp[i])) {
                                intg[i]+=-deadup[i]-output[i];
                                flag[i]=0;
                        }
                } else flag[i]=1;
                if(i==2) uboost=1000.0*ub;
        /* Spurious comment */
        }
        output[1]=0.0;
        output[3]=0.0;
        return 1;
}

int equiv_f_cont(float *state,float *state_d, float h, float *output)
{
        float minerr[4]={0.0,0.03,0.03,0.00105};
        static float intg[4]={0.0,0.0,0.0,0.0};
        static flag[4]={1,1,1,1};
        float boostthresh[4]={0.0,0.15,0.15,0.0015};
        float deadcomp[4]={0.0,450.0,450.0,650.0};
        float deaddown[4]={0.0,500.0,500.0,800.0};
        float deadup[4]={0.0,500.0,650.0,1000.0};
        float qddotmax[4]={0.0,10.0,10.0,0.2};
        float err,vel,nonlin,a,ub;
        int i=2;
        void mod_f(float *fin, float *fout);
        void mod_c(float *uin, float *uout);
        float f_mod[4];
        static float fo[4]={0.0,0.0,0.0,0.0};
        static float fo2[4]={0.0,0.0,0.0,0.0};

        mod_f(state_d,f_mod);

        err = f_mod[3]-(state[3]+fo[3]+fo2[3])/3.0;

        intg[i] += Ki[i]/4.0*err*h;
        output[i] = Kp[i]*err + intg[i];
                mod_c(output,output);
        output[i]*=rtd;

        fo2[3]=fo[3];
        fo[3]=state[3];

        output[1]=0.0;
        output[3]=0.0;
        return 1;
}
/*****************************************************************************/
float anti_stick_boost(float v, float v_set)
{
        float b=50.0, nonlin;

        nonlin = v_set*v_set;
        nonlin = nonlin/(nonlin + b*v*v + 0.0001);  // epsilon for div by 0
        return (v_set-v)*nonlin;
}
/*****************************************************************************/
void set_qd(float *q_d, float time)
{
        float delxx[4],delxf[4],x_des[4],x_desired[4];
        float x_desired_less_end[4], f_desired_less_end[4];
        float factor;
        int i;
        void xtoq(float *x, float *q);
        void second_order_model_f(float* delxf,float* fend);
        void second_order_model_x(float* delxx,float* x_desired);

       x_desired[1]=xi[1];
```

```
        x_desired[2]=xi[2];
        x_desired[3]=xi[3];

        fd[1]=0.0;
        fd[2]=0.0;
        fd[3]=0.0;
        if(time>2.0) fd[3]+=100.0;

//      fd[3]*=k*(1.0/env + 1.0/k);   // Force set-point compensation

        for(i=1;i<=3;i++) {
           x_desired_less_end[i]=x_desired[i]-x[i];
           f_desired_less_end[i]=fend[i]-fd[i];
        }
//      fd[3]/=k*(1.0/env + 1.0/k);   // Force set-point reset

        second_order_model_f(delxf,f_desired_less_end);
        second_order_model_x(delxx,x_desired);
        for(i=1;i<=3;i++) {
              x_des[i]=delxx[i]+delxf[i];
        }
        xtoq(x_des,q_d);
}
/* ********************************************************************** */
void set_parameters()
{
  x=vector(1,3);
  xd=vector(1,3);
  xi=vector(1,3);
  qd=vector(1,3);
  qddot=vector(1,3);
  qddot[1]=0.0;
  qddot[2]=0.0;
  qddot[3]=0.0;
  q=vector(1,3);
  qi=vector(1,3);
  qdot=vector(1,3);
  qdot[1]=0.0;
  qdot[2]=0.0;
  qdot[3]=0.0;
  u=vector(1,3);
  fd=vector(1,3);
  u[1]=0.0;
  u[2]=0.0;
  u[3]=0.0;
  HighLimit=vector(1,3);
  HighLimit[1]=110.0/rtd;
  HighLimit[2]=27.0/rtd;
  HighLimit[3]=1.828; //1.05
  LowLimit=vector(1,3);
  LowLimit[1]=-110.0/rtd;
  LowLimit[2]=-30.0/rtd;
  LowLimit[3]=0.788;  //0.0
  fend=vector(1,3);
  fend[1]=0.0;
  fend[2]=0.0;
  fend[3]=0.0;
  alpha=vector(1,3);
  alpha[1]=5000.0;
  alpha[2]=8000.0;
  alpha[3]=20.0;
  Ki=vector(1,3);
  Ki[1]=8e5/rtd;
  Ki[2]=15e4/rtd;
  Ki[3]=15e5;
  Ka=vector(1,3);
  Ka[1]=150.0;
  Ka[2]=150.0;
  Ka[3]=20000.0;
  Kp=vector(1,3);
  Kp[1]=3e4/rtd;
  Kp[2]=7e4/rtd;
  Kp[3]=1e5;
  kenv=6200.0;
  k=50.0;
```

```
   k_i=k;
   m=500.0;
   wn=sqrt(k/m);
   eta_net = 3.0;
   c=2.0*eta_net*sqrt((k+kenv)*m);
   eta=c/(2.0*m*wn);
}
/* ********************************************************************** */
void mod_f(float *fin, float *fout)          // for equivalence analysis
{
        static float xo[4]={0.0,0.0,0.0,0.0};
        static float vo[4]={0.0,0.0,0.0,0.0};
        float wd,wn=sqrt((k+kenv)/m),eta=eta_net;
        float et,st,ct,a,b;
        int i;

        if(eta>1.0) {wd=wn*sqrt(eta*eta-1.0);}
        else {wd=wn*sqrt(1.0-eta*eta);}

        et=exp(-deltat*wn*eta);

        for(i=1;i<=3;i++) {

        a=xo[i]-fin[i]/(k+kenv);
        b=vo[i]+eta*wn*a;
        if(eta>1.0) {
                st=exp(wd*deltat);
                ct=(st+1./st)/2.0;
                st=(st-1./st)/2.0;
                fout[i]=et*(a*ct+b*st/wd);
                vo[i]=-eta*wn*fout[i] + et*(wd*a*st+b*ct);
        } else if (eta==1.0) {
                fout[i]=et*(a+b*deltat);
                vo[i]=-eta*wn*fout[i]+b*et;
        } else {
                st=sin(wd*deltat);
                ct=cos(wd*deltat);
                fout[i]=et*(a*ct+b*st/wd);
                vo[i]=-eta*wn*fout[i] + et*(-wd*a*st+b*ct);
        }
        fout[i]+=xo[i]-a;
        xo[i]=fout[i];
        fout[i]*=kenv;
        }
}
/* ********************************************************************** */
void mod_c(float *cin, float *cout)          // for equivalence analysis
{
        static float xo[4]={0.0,0.0,0.0,0.0};
        static float vo[4]={0.0,0.0,0.0,0.0};
        static float to[4]={0.0,0.0,0.0,0.0};
        float wd;
        float et,st,ct,a,b,tmp1,tmp2;
        int i;

        if(eta>1.0) {wd=wn*sqrt(eta*eta-1.0);}
        else {wd=wn*sqrt(1.0-eta*eta);}

        et=exp(-deltat*wn*eta);

        for(i=1;i<=3;i++) {

       a=xo[i]-cin[i]/k;
       b=vo[i]+eta*wn*a;
       if(eta>1.0) {
               st=exp(wd*deltat);
               ct=(st+1./st)/2.0;
               st=(st-1./st)/2.0;
               cout[i]=et*(a*ct+b*st/wd);
               vo[i]=-eta*wn*cout[i] + et*(wd*a*st+b*ct);
       } else if (eta==1.0) {
               cout[i]=et*(a+b*deltat);
               vo[i]=-eta*wn*cout[i]+b*et;
       } else {
               st=sin(wd*deltat);
```

```
                ct=cos(wd*deltat);
                cout[i]=et*(a*ct+b*st/wd);
                vo[i]=-eta*wn*cout[i] + et*(-wd*a*st+b*ct);
        }
        cout[i]+=xo[i]-a;
        tmp1=m*(cout[i]-xo[i])/deltat;
        tmp2=(c*(cout[i]-xo[i]) + tmp1-to[i])/deltat;
        to[i]=tmp1;
        xo[i]=cout[i];
        cout[i]=(tmp2+(k+kenv)*cout[i])/kenv;
        }
}
/* ********************************************************************** */
void second_order_model_f(float *delxf, float *fend)
{
        static float xo[4]={0.0,0.0,0.0,0.0};
        static float vo[4]={0.0,0.0,0.0,0.0};
        static float fo[4]={0.0,0.0,0.0,0.0};
        static float fo2[4]={0.0,0.0,0.0,0.0};
        static float wd;
        float et,st,ct,a,b;
        int i;

        if(eta>1.0) {wd=wn*sqrt(eta*eta-1.0);}
        else {wd=wn*sqrt(1.0-eta*eta);}

        et=exp(-deltat*wn*eta);

        for(i=1;i<=3;i++) {

        a=xo[i]-(fend[i]+fo[i]+fo2[i])/3.0/k;
        b=vo[i]+eta*wn*a;
        if(eta>1.0) {
                st=exp(wd*deltat);
                ct=(st+1./st)/2.0;
                st=(st-1./st)/2.0;
                delxf[i]=et*(a*ct+b*st/wd);
                vo[i]=-eta*wn*delxf[i] + et*(wd*a*st+b*ct);
        } else if (eta==1.0) {
                delxf[i]=et*(a+b*deltat);
                vo[i]=-eta*wn*delxf[i]+b*et;
        } else {
                st=sin(wd*deltat);
                ct=cos(wd*deltat);
                delxf[i]=et*(a*ct+b*st/wd);
                vo[i]=-eta*wn*delxf[i] + et*(-wd*a*st+b*ct);
        }
        delxf[i]+=xo[i]-a;
        fo2[i]=fo[i];
        fo[i]=fend[i];
        xo[i]=delxf[i];
        }
}
/* ********************************************************************** */
void second_order_model_x(float *delxx, float *x_desired)
{
        static float xo[4]={0.0,0.0,0.0,0.0};
        static float vo[4]={0.0,0.0,0.0,0.0};
        static float xdo[4]={0.0,0.0,0.0,0.0};
        static float wd;
        static int flag=1;
        float et,st,ct,a,b;
        int i;

        if(flag) {
                for(i=1;i<=3;i++) {
                        xo[i]=xi[i];
                        xdo[i]=xi[i];
                }
                flag=0;
        }

        if(eta>1.0) {wd=wn*sqrt(eta*eta-1.0);}
        else {wd=wn*sqrt(1.0-eta*eta);}
```

```
        et=exp(-deltat*wn*eta);

        for(i=1;i<=3;i++) {

        a=xo[i]-x_desired[i];
        b=vo[i]+eta*wn*(a+2*(x_desired[i]-xdo[i]));
        if(eta>1.0) {
                st=exp(wd*deltat);
                ct=(st+1./st)/2.0;
                st=(st-1./st)/2.0;
                delxx[i]=et*(a*ct+b*st/wd);
                vo[i]=-eta*wn*delxx[i] + et*(wd*a*st+b*ct);
        } else if (eta==1.0) {
                delxx[i]=et*(a+b*deltat);
                vo[i]=-eta*wn*delxx[i]+b*et;
        } else {
                st=sin(wd*deltat);
                ct=cos(wd*deltat);
                delxx[i]=et*(a*ct+b*st/wd);
                vo[i]=-eta*wn*delxx[i] + et*(-wd*a*st+b*ct);
        }
        delxx[i]+=xo[i]-a;
        xdo[i]=x_desired[i];
        xo[i]=delxx[i];
        }
}
/* ********************************************************************** */
void qtox(float *q, float *x)
{
        float s2,c2,temp,a2prime,d3prime;

        s2=sin(q[2]/rtd);
        c2=cos(q[2]/rtd);
        d3prime=q[3]+0.788-a4*sin(q4/rtd);
        a2prime=a2-a4*cos(q4/rtd);
        temp=d3prime*c2+a2prime*s2;
        x[1]=cos(q[1]/rtd)*temp;
        x[2]=sin(q[1]/rtd)*temp;
        x[3]=-d3prime*s2+a2prime*c2;
}
/* ********************************************************************** */
void xtoq(float *x, float *q)
{
        float temp,a2prime,d3prime,limit,dplus,aplus;

        dplus=-a4*sin(q4/rtd);
        aplus=-a4*cos(q4/rtd);
        a2prime=a2+aplus;
        temp=sqrt(x[1]*x[1]+x[2]*x[2]);

        /* Check that set-point is roughly within operating envelope */

        limit=LowLimit[3]+dplus;
        if (temp<limit) {
                x[1]/=temp/limit;
                x[2]/=temp/limit;
                temp=limit;
                printf("Modifying set-point:  Lower limit on d3\n");
        } else if (temp>(limit=(HighLimit[3]+dplus)*cos(HighLimit[2]))) {
                x[1]/=temp/limit;
                x[2]/=temp/limit;
                temp=limit;
                printf("Modifying set-point:  Upper limit on d3\n");
        }
        limit=(HighLimit[3]+dplus)*sin(-HighLimit[2])+aplus*cos(-HighLimit[2]);
        if (x[3]<limit) {
                printf("Modifying set-point:  Z too low\n");
                x[3]=limit;
        } else if (x[3]>(limit=(HighLimit[3]+dplus)*sin(-LowLimit[2])
                                +aplus*cos(-LowLimit[2])))  {
                printf("Modifying set-point:  Z too high\n");
                x[3]=limit;
        }
        q[1]=atan2(x[2],x[1])*rtd;
        d3prime=sqrt(temp*temp+x[3]*x[3]-a2prime*a2prime);
```

```
        q[2]=atan2(-(d3prime*x[3]-a2prime*temp),
                   (d3prime*temp+a2prime*x[3]))*rtd;
        q[3]=d3prime-dplus-0.788;
}
/* ************************************************************************ */
void get_foffset(void)
{
        int i=0,value;
        float fsum=0.0;

                clrscr();
        printf("Measuring force sensor bias..\n");
        while (i++<500) {
                value = get_a_to_d(15);
                fsum+=value*20.0/66.0*9.8/2.2;  // Newtons (I think)
        }
        foffset=fsum/i;
}

/* ************************************************************************ */
void nrerror(char *error_text)
{
        fprintf(stderr,"Numerical Recipes Run-time error...\n");
        fprintf(stderr,"%s\n",error_text) ;
        fprintf(stderr,"...now exiting to system...\n") ;
        exit(1) ;
}
/* ************************************************************************ */
void free_vector(float *v,long nl,long nh)
{
        free((char*) (v+nl));
}

/* ************************************************************************ */
float *vector(long nl,long nh)
{
    float *v ;

         v=(float *)malloc((unsigned) (nh-nl+1)*sizeof(float));
    if (!v) nrerror("allocaton failure in vector ()");
         return v-nl;
}
/*************************************************************************/
trigger_switch()
        {
        static int d_bit=1,flag=1;
        if ((inportb (A_TO_D_BASE_ADDRESS + 3) & 1) == 1) flag = 1 ;
        if ((inportb (A_TO_D_BASE_ADDRESS + 3) & 1) == 0)
                {
                if(flag == 1)
                        {
                        if(d_bit == 1) d_bit = 0;
                  else d_bit = 1;
                        outportb(D_TO_A_BASE_ADDRESS + 14 , d_bit) ;
                        flag = 0;
                        gotoxy(10,20);
                        if(d_bit ==1)printf("CLOSED");
                  else printf("OPEN  ");

                        }
                }
        }

write_information_to_monitor()
        {
        clrscr();
        puts(" RESPONSE IS 60% OF MAXIMUM FOR AMATURE OPERATION!");
        puts("++++++++++ joystick inputs ++++++++++++++++++++++++");
        puts("          X           Y           Z           W           U           R");
        gotoxy(1,5);
        puts("++++++++++ VALVE OUTPUTS ++++++++++++++++++++++++++");
        puts("          X           Y           Z           W           U ");
        gotoxy(1,8);
        puts("--------------- pressures ------------------------------");
        puts("           rot_left   rot_right   extend   retract      up      down");
```

```
        puts("  DIGITAL:");
        puts(" 0.1 VOLT:");
        puts("FORCE(LB):");
        puts("NET FORCE:");


        gotoxy(1,14);
        puts("++++++++++ JOINT COORDINATES ++++++++++++++++++++++");
        puts("          X           Y           Z           W           U");
        puts("       ROTATE      IN/OUT      UP/DOWN       ROLL       PITCH");
        puts("\nForce gauge=      digital, =       Volts, =         pounds");
        gotoxy(1,24);
         while (kbhit() ) getch();    /* clear keyboard buffer */
        puts("------- PRESS ANY KEY TO *** START *** THIS PROGRAM");
        getch();
        gotoxy(1,24);
        puts("PRESS ANY KEY TO STOP THIS PROGRAM.................");
         t1 = readtimer();
        }


 void initialize_d_to_a()
        {
        outportb(D_TO_A_BASE_ADDRESS + 15, 0x92); /* makes port an output port */
        outportb(D_TO_A_BASE_ADDRESS + 14 , 1  ); /* WRITE INITIAL CLOSE COMMAND*/
        }

 void initialize_pio12()
        {
        outportb (PIO12_BASE_ADDRESS+3 ,128+18);    /* makes port c(lower) output while
 ports A and B are inputs*/
        }

 int get_a_to_d(int a_to_d_channel)
 {
        int input_value;

        outportb(A_TO_D_BASE_ADDRESS + 2, a_to_d_channel);/* set channel*/
        delay(1);

        outportb(A_TO_D_BASE_ADDRESS, 0);
        /*cause conversion on chosen channel */

        while( (inportb(A_TO_D_BASE_ADDRESS + 8) & 0x80) != 0);
        /* wait for completed conversion */

        input_value =inportb(A_TO_D_BASE_ADDRESS)/16 +
                        inportb(A_TO_D_BASE_ADDRESS+1)*16 - 2048;

        if(abs(input_value)>0)  /* 40 is a good dead band */
        return(input_value);
        else  return(0);
 }

 void zero_voltage_and_current_outputs()
 {
        data = 2048;
        for (d_to_a_channel =0; d_to_a_channel < 6;
        d_to_a_channel++) out_d_to_a (data);
 }

 void out_d_to_a(data)
 {
        int data_high,data_low;

        if (data > 4095) data = 4095;
        if (data < 0    ) data = 0;

        data = (data -2048)*3/5+2048; /* reduce response for amature operator */
        data_high = (data/256);
        data_low  = data - data_high * 256;
        outportb(D_TO_A_BASE_ADDRESS + 2*d_to_a_channel, data_low);
        outportb(D_TO_A_BASE_ADDRESS + 2*d_to_a_channel +1 , data_high);
 }
```

```
void initializetimer(void)
{
  outportb(0x043,0x034);
  asm jmp short NullJump1
NullJump1:;
  outportb(0x040,0x000);
  asm jmp short NullJump2
NullJump2:;
  outportb(0x040,0x000);
}

void restoretimer(void)
{
  outportb(0x043,0x036);
  asm jmp short NullJump1
NullJump1:;
  outportb(0x040,0x000);
  asm jmp short NullJump2
NullJump2:;
  outportb(0x040,0x000);
}

long readtimer(void)
 {
  asm cli              /* Disable interrupts */
  asm mov  dx,020h     /* Address PIC ocw3   */
  asm mov  al,00Ah     /* Ask to read irr    */
  asm out  dx,al
  asm mov  al,00h      /* Latch timer 0 */
  asm out  043h,al
  asm in   al,dx       /* Read irr      */
  asm mov  di,ax       /* Save it in DI */
  asm in   al,040h           /* Counter --> bx*/
  asm mov  bl,al       /* LSB in BL     */
  asm in   al,040h
  asm mov  bh,al       /* MSB in BH     */
  asm not  bx          /* Need ascending counter */
  asm in   al,021h     /* Read PIC imr  */
  asm mov  si,ax       /* Save it in SI */
  asm mov  al,00FFh   /* Mask all interrupts */
  asm out  021h,al
  asm mov  ax,040h     /* read low word of time */
  asm mov  es,ax       /* from BIOS data area   */
  asm mov  dx,es:[06Ch]
  asm mov  ax,si       /* Restore imr from SI   */
  asm out  021h,al
  asm sti              /* Enable interrupts */
  asm mov  ax,di       /* Retrieve old irr  */
  asm test al,001h     /* Counter hit 0?    */
  asm jz   done        /* Jump if not       */
  asm cmp  bx,0FFh     /* Counter > 0x0FF?    */
  asm ja   done        /* Done if so        */
  asm inc  dx          /* Else count int req. */
  done:;
  asm mov ax,bx        /* set function result */
 }

/* ********************************************************************** */
void regress_velocity_vector(double time, float *x, float *v, int i)
/*
       x[1->3] position vector
       v[1->3] output velocity vector
       0<=i<MAX_REG quantity index, allowing different concurrent regressions
*/
{
       int j,k;
       static double t[N_reg][MAX_REG],ang[N_reg][3][MAX_REG];
       static double sumx2[3][MAX_REG],sumtx[3][MAX_REG],sumx[3][MAX_REG],
                     sumt[MAX_REG],sumt2[MAX_REG];
       static int f=1,flag[MAX_REG],p[MAX_REG];

       /* Velocity initialization */

       if(i>=MAX_REG) return;
       if(f) {
```

```
                for(j=0;j<MAX_REG;j++) {
                        flag[j]=1;
                        p[j]=0;
                }
                f=0;
        }
        if(flag[i]) {
                for(k=0;k<3;k++) {
                        for(j=0;j<N_reg;j++) {
                                t[j][i]=time*(1-0.04*(N_reg-j));
                                ang[j][k][i]=x[k+1];
                        }
                        sumx[k][i]=0.0;
                        sumx2[k][i]=0.0;
                        sumtx[k][i]=0.0;
                        sumt[i]=0.0;
                        sumt2[i]=0.0;
                        v[k+1]=0.0;
                        t[1][i]=0.0;
                        for(j=0;j<N_reg;j++) {
                                sumx[k][i]  += ang[j][k][i];
                                sumx2[k][i] += ang[j][k][i] * ang[j][k][i];
                                sumtx[k][i] += ang[j][k][i] * t[j][i];
                                sumt[i]  += t[j][i];
                                sumt2[i] += t[j][i] * t[j][i];
                        }
                }
                flag[i]=0;
        }

        /* Update sums */
        sumt[i]  += time - t[p[i]][i];
        sumt2[i] += time*time - t[p[i]][i]*t[p[i]][i];

        for(k=0;k<3;k++) {

                /* Update sums */
                sumx[k][i]  += x[k+1] - ang[p[i]][k][i];
                sumx2[k][i] += x[k+1]*x[k+1] - ang[p[i]][k][i]*ang[p[i]][k][i];
                sumtx[k][i] += x[k+1]*time - ang[p[i]][k][i]*t[p[i]][i];

                /* Store current value over oldest value */
                ang[p[i]][k][i] = x[k+1];

                /*******************************/
                /* Actual velocity calculation */
                /*******************************/

                if ((N_reg*sumt2[i]-(sumt[i]*sumt[i])) != 0.0 && t[p[i]+1][i] != 0.0)
                     { v[k+1] = (float)((N_reg*sumtx[k][i] - sumt[i]*sumx[k][i])
                                /(N_reg*sumt2[i] - sumt[i]*sumt[i]));}
                else { v[k+1] = 0.0; }
                gotoxy(1,18);
                }
        /* Store current value over oldest and increment pointer */
        t[p[i]][i] = time;
        p[i]+=1;
        if(p[i]>=N_reg) p[i]=0;
}

void store_data(int i)
{
   data_block[IND_t][i]=time;
   data_block[IND_1][i]=q[1];
   data_block[IND_1d][i]=qd[1];
   data_block[IND_2][i]=q[2];
   data_block[IND_2d][i]=qd[2];
   data_block[IND_3][i]=q[3];
   data_block[IND_3d][i]=qd[3];
   data_block[IND_u2][i]=u[2];
   data_block[IND_force][i]=fend[3];
   data_block[IND_fd][i]=fd[3];
   data_block[IND_k][i]=k;
}
```

```
void save_data()
{
   float fi=0.0,offset[4];
   int i=0,j;

   printf("Saving %d data points, spaced by %f seconds...\n",ndata,DTSAVE);
   open_files(savedir);
   q[1]=data_block[IND_1][i];
   q[2]=data_block[IND_2][i];
   q[3]=data_block[IND_3][i];
   qtox(q,x);
   offset[1]=0.0+xi[1];
   offset[2]=0.0+xi[2];
   offset[3]=0.0+xi[3];
   for(i=0;i<ndata;i++) {
        q[1]=data_block[IND_1][i];
        q[2]=data_block[IND_2][i];
        q[3]=data_block[IND_3][i];
        qtox(q,x);
        qd[1]=data_block[IND_1d][i];
        qd[2]=data_block[IND_2d][i];
        qd[3]=data_block[IND_3d][i];
        qtox(qd,xd);
        fprintf(fout,"%f\t%f\n",data_block[IND_t][i],data_block[IND_force][i]);
        fprintf(fdout,"%f\t%f\n",data_block[IND_t][i],data_block[IND_fd][i]);
        fprintf(zout,"%f\t%f\n",data_block[IND_t][i],x[3]-offset[3]);
        fprintf(zdout,"%f\t%f\n",data_block[IND_t][i],xd[3]-offset[3]);
        fprintf(xlout,"%f\t%f\t%f\t%f\t%f\t%f\n",data_block[IND_t][i],
               data_block[IND_force][i],data_block[IND_fd][i],
               x[3]-offset[3],xd[3]-offset[3],
               data_block[IND_u2][i]);
   }
close_files();
}

void open_files(char *dir)
{
   char string[60];

   strcat(dir,"\\");
   strcpy(string,dir);
   strcat(string,"f.exp");
   fout=openfile(string,"w");
   strcpy(string,dir);
   strcat(string,"fd.exp");
   fdout=openfile(string,"w");
   strcpy(string,dir);
   strcat(string,"z.exp");
   zout=openfile(string,"w");
   strcpy(string,dir);
   strcat(string,"zd.exp");
   zdout=openfile(string,"w");
   strcpy(string,dir);
   strcat(string,"xl.exp");
   xlout=openfile(string,"w");
}

FILE* openfile(char *filename, char *mode)
{
       FILE *fp;
       if((fp=fopen(filename, mode))==NULL) {
              printf("Error openning file %s for %s\n",filename,mode);
              exit(1);
       } else { return fp; }
}

void close_files(void)
{
       fclose(fout);
       fclose(fdout);
       fclose(zout);
       fclose(zdout);
       fclose(xlout);
}
```

```
/* ********************************************************************** */

void path(float *q_des,float *qdot_des,float pos_min,float pos_max,float T[],
              float t_traj[],float t_acc,float time)
/*  T = vector of times at constant velocity (v+, 0, v-)
    t_traj = times in different portions of trajectory
    t_acc = (T[0]-t_traj[0])/2, but have 4*t_acc per segment */
{
    float t_rel;
    if(time<=t_traj[0])
        trajectory(pos_min,pos_max,T[0],t_acc,time,q_des,qdot_des);
    if(time>t_traj[0] && time<=t_traj[0]+t_traj[1]){
        time-=t_traj[0];
        trajectory(pos_max,pos_max,T[1],t_acc,time,q_des,qdot_des);
    }
    if(time>t_traj[0]+t_traj[1] && time<=t_traj[0]+t_traj[1]+t_traj[2]){
        time-=(t_traj[0]+t_traj[1]);
        trajectory(pos_max,pos_min,T[2],t_acc,time,q_des,qdot_des);
    }
    if(time>t_traj[0]+t_traj[1]+t_traj[2]) *q_des=*qdot_des=0.0;
}

void trajectory(float qi,float qf,float T,float t_acc,float t_rel,float *q_des,
              float *qdot_des)
{
    float A,B,C,dB,dC,h,tiny;
    if(t_rel<T) A=B=qi;
    else{
        h=1.0-t_acc/T;
        A=(qf-qi)*h+qi;
        B=qf;
    }
    C=qf;
    dB=A-B;
    dC=C-B;
    if(t_rel<2.0*t_acc){
        h=t_rel/(2.0*t_acc);
        *q_des=((dC*t_acc/T+dB)*(2.0-h)*h*h-2.0*dB)*h+A;
        *qdot_des=((dC*t_acc/T+dB)*(1.5-h)*2.0*h*h-dB)/t_acc;
    }
    if(t_rel>=2.0*t_acc && t_rel<T){
        h=(t_rel-t_acc)/T;
        *q_des=dC*h+B;
        *qdot_des=dC/T;
    }
    if(t_rel>=T){
        h=(t_rel-T)/(2.0*t_acc);
        *q_des=((dC*t_acc/T+dB)*(2.0-h)*h*h-2.0*dB)*h+A;
        *qdot_des=((dC*t_acc/T+dB)*(1.5-h)*2.0*h*h-dB)/t_acc;
    }
}
```