

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

Robust Torque Control of Harmonic Drive Systems

Hamid D. Taghirad

Department of Electrical Engineering
McGill University, Montréal

May 1997

A Thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements of the degree of
Doctor of Philosophy

© HAMID D. TAGHIRAD, 1997



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-44660-3

Canada

To my dear wife
and
my beloved mother

ABSTRACT

A harmonic drive is a compact, light-weight and high-ratio torque transmission device which has almost zero backlash. Its unique performance features has captured the attention of designers in many industrial applications, especially in robotics. Despite widespread industrial application of harmonic drives, the torque control of this system has not been fully addressed. In this thesis the robust torque control of harmonic drive system is examined in detail.

In order to measure the transmitted torque of a harmonic drive and for the purpose of a torque feedback scheme, an intelligent built-in torque sensing technique is developed in this thesis. Specifically, strain-gauges are mounted directly on the flexspline and therefore no extra flexible element is introduced into the system. To have maximum sensing accuracy, four Rosette strain gauges are employed using an accurate positioning method. Kalman filter estimation is employed to cancel the torque ripples, oscillations observed on the measured torque and caused mainly by gear teeth meshing. A simple forth order harmonic oscillator proved sufficient to model these torque ripples. Moreover, the error model is extended to incorporate any misalignment torque. By on-line implementation of the Kalman filter, it is shown that this method affords a fast and accurate way to filter torque ripples and misalignment torque from torque measurements.

Based on experimental and theoretical studies, a systematic way to capture and rationalize the dynamic behavior of the harmonic drive systems is developed next in this thesis. Simple and accurate models for compliance, hysteresis, and friction are proposed, and the model parameters are estimated using least-squares approximation for linear and nonlinear regression models. A statistical measure of variation is defined, by which the reliability of the estimated parameter for different operating condition, as well as the accuracy and integrity of the proposed model is quantified. By these means, it is shown that a linear stiffness model best captures the behavior of the system when combined with a good model for hysteresis. Moreover, the frictional losses of harmonic drive are modelled at both low

and high velocities. The model performance is assessed by comparing simulations with the experimental results on two different harmonic drives.

Finally, robust $\mathcal{H}_\infty$ -based torque controllers are designed and implemented for harmonic drive systems under constrained- and free-motion applications. A nominal model for the system is obtained in each case from ensembles of experimental frequency response estimates of the system, while the deviation of the system from the model is encapsulated by a multiplicative uncertainty. Robust $\mathcal{H}_\infty$ -based torque controllers are designed using this information, and the controllers are implemented on two different setups using the Kalman filtered torque as an integral part of the torque-feedback loop. Exceptional performance results are obtained from the time and frequency response of the closed-loop system, especially for the constrained-motion system. To further improve the performance of the free-motion system, a model-based friction-compensation algorithm is implemented in addition to the robust torque control. It is shown that the friction-compensation shrinks the model uncertainty at low frequencies. Hence, the performance of the closed-loop system is improved for tracking signals with low-frequency content.

RÉSUMÉ

Un entraînement harmonique est un mécanisme léger, de rapport de transmission élevé et qui n'a pratiquement pas de jeu. Ses performances uniques ont attiré l'attention de concepteurs dans diverses applications industrielles, particulièrement en robotique. Cependant, dans l'industrie malgré la prépondérance d'entraînements harmoniques la commande du couple du système n'a pas encore fait l'objet d'étude complète. Dans cette thèse nous examinons la commande de couple robuste des systèmes à entraînement harmonique en détail.

Pour mesurer et utiliser le couple transmis par un entraînement harmonique en boucle fermée, nous développons une technique intelligente de détection du couple. Dans celle-ci on monte des extensomètres directement sur l'élément flexible, et donc il n'y a pas d'autre élément flexible introduit dans le système. Afin de minimiser les erreurs de lecture, nous employons quatre jauges de contrainte utilisant une méthode précise de positionnement. Un filtre de Kalman est utilisé pour réduire les ondulations du couple. Les oscillations observées sur le couple mesuré sont dues principalement à l'interférence des dents l'engrenage. Un oscillateur harmonique du quatrième ordre a suffit pour modéliser de façon précise les ondulations du couple. De plus, l'usage du filtre de Kalman montre que cette méthode constitue un moyen à la fois rapide et précis de filtration des ondulations du couple et des erreurs d'alignement dues aux mesures.

En nous basant sur des études expérimentales et théoriques, nous proposons une étude systématique et rationnelle du comportement dynamique de l'entraînement harmonique. Des modèles simples et précis pour l'élasticité, l'hystérésis et le frottement sont proposés, et les paramètres du modèle sont estimés par la méthode des approximations des moindres carrés pour modèles linéaire et non-linéaire régressifs. Nous définissons une mesure statistique de variation avec laquelle on quantifie non seulement la fiabilité des paramètres estimés pour différentes conditions d'opération mais aussi la précision et l'intégrité du modèle proposé. Par ce moyen, on montre qu'un modèle linéaire de la raideur permet une meilleur

reproduction du comportement du système lorsqu'il est combiné avec un modèle précis de l'hystérésis. De plus, les pertes dues aux frottements sont modélisées à basses et grandes vitesses. Les performances du modèle sont illustrées par simulation, vérifiant ainsi les résultats expérimentaux sur deux entraînements harmoniques différents.

Finalement, nous faisons la conception et l'implémentation de contrôleurs $\mathcal{H}_\infty$ -robuste du couple pour des systèmes à entraînements harmoniques avec mouvements libre et contraint. Un modèle nominal du système est obtenu dans chaque cas à partir des réponses fréquentielles du système, et la déviation de ce dernier du modèle est réduite à une incertitude multiplicative. Les contrôleurs $\mathcal{H}_\infty$ sont conçus en utilisant cette information, et mis en oeuvre sur deux dispositifs utilisant le couple filtré comme partie intégrale de la rétroaction couple. Des performances et résultats exceptionnels sont obtenus des réponses temporelles et fréquentielles du système en boucle fermé. Pour améliorer davantage la performance du système en mouvement libre, on ajoute un algorithme de compensation du frottement à la commande robuste du couple. On démontre ainsi que la compensation du frottement réduit l'incertitude aux basses fréquences, et par conséquent la performance du système en boucle fermée s'en trouve améliorée à ces mêmes fréquences.

ACKNOWLEDGEMENTS

I wish to extend my sincere gratitude and appreciation to my research advisor, Dr. Pierre R. Bélanger, for his superb guidance throughout my research studies. His penetrating insight and rigour were crucial to the successful completion of this thesis.

I enjoyed attending several lectures and seminars in the Departments of Electrical and Mechanical Engineering and in the Centre for Intelligent Machines, and my sincere gratitude goes to the whole faculty and staff. Special thanks to Drs. George Zames and Vincent Hayward for many discussions throughout the course of this research. I would like to express my appreciation to A. Helmy for his involvements in the experiments as well as and my fellow students at the Centre for Intelligent Machines, especially Kaouthar Benameur and Mojtaba Ahmadi for many interesting and enjoyable discussions. Special thanks to my officemates Michael Glaum for his careful review of the manuscript, and Mohammed Seddik Djouadi for his assistance in the French translation of the abstract. I am grateful to the machine shop staff of the Department of Electrical Engineering, especially Don Pavlasek for his valuable suggestions and assistance toward developing the torque sensor and the harmonic drive testing stations. The assistance of the Photonic System Group of Electrical Engineering Department for equipment employed in the development of the torque sensor is gratefully acknowledged.

The support of this research by the Natural Science and Engineering Council of Canada (NSERC), the Institute for Robotics and Intelligent Systems (IRIS), the Faculty of Graduate Studies and Research of McGill University, and the Ministry of Culture and Higher Education (MCHE) of the Islamic Republic of Iran are gratefully acknowledged. Some of the equipment used in this research was gratefully provided by CAE Electronics Inc. and International Submarine Engineering Ltd.

I would like to express my appreciation to my wife for her boundless support and patience, and I praise God for his continuous support while accomplishing this research.

CLAIMS OF ORIGINALITY

In this thesis the following original contributions are presented:

- An accurate strain–gauge positioning method is proposed for harmonic drive built–in torque sensor.
- A Kalman filter estimation routine is proposed to filter torque ripples and misalignment torque signatures from measured torque.
- A Simple and accurate model for compliance is proposed which can capture the hysteresis behaviour of the harmonic drive system.
- A strategy for estimating friction losses of the harmonic drive system is given for both low– and high–velocities.
- The proposed modelling and identification scheme of the harmonic drive system is verified by experiments.
- The nonlinear dynamics of harmonic drive systems is encapsulated by a linear model and an uncertainty representation, using empirical frequency response estimates of the system.
- Robust $\mathcal{H}_\infty$ –based torque controllers are designed and implemented for both free– and constrained–motion systems.
- A model–based friction–compensation algorithm is designed and implemented on the system in addition to the $\mathcal{H}_\infty$ controller.

TABLE OF CONTENTS

ABSTRACT	ii
RÉSUMÉ	iv
ACKNOWLEDGEMENTS	vi
CLAIMS OF ORIGINALITY	vii
LIST OF FIGURES	xii
LIST OF TABLES	xvii
CHAPTER 1. Introduction	1
1. Harmonic Drive Systems	1
1.1. Principles of Operation	2
1.2. Performance Features	3
2. Research Objectives	4
3. Literature Survey	6
3.1. The Technological Status	6
3.2. Torque Sensing	6
3.3. Modelling and Identification	7
3.4. Control	8
4. Thesis Outline	9
CHAPTER 2. The Experimental Setup Hardware & Software	11
1. Harmonic Drive Testing Stations	11
1.1. McGill Setup	12
1.2. ISE Setup	16
2. The Siglab Hardware	18
2.1. Virtual Sine-Sweep (VSS)	18

2.2. Virtual Network Analyzer (VNA)	20
2.3. System Identification	20
3. Data Acquisition and Processing Hardware	20
3.1. The Challenger-C30V Specifications	21
3.2. Other I/O boards' Specifications	22
3.3. Modified Hardware for Frequency Response Experiments	23
4. Software	25
4.1. Time Response Software	25
4.2. Frequency Response Software	29
CHAPTER 3. Intelligent Built-in Torque Sensor	31
1. Built-In Torque Sensor	32
1.1. Positioning Error of Strain-gauges	34
2. Torque Sensor Calibration	35
3. Torque Ripple Compensation	37
3.1. Torque Ripple Model	38
3.2. Torque Ripple Estimation	39
4. Misalignment Torque Compensation	40
5. Summary	42
CHAPTER 4. Modelling and Identification	44
1. DC Motor	44
2. Servo-Amplifier	46
3. Harmonic Drive	48
3.1. Harmonic Drive Compliance	49
3.2. Harmonic Drive Friction	52
4. Summary	55
CHAPTER 5. Simulations	57
1. Modelling and Identification Verification	57
1.1. System under Constrained-Motion	57
1.2. System under Free-Motion	60
1.3. Discussions	61
2. System Simulations	62
2.1. System under Constrained-Motion	62
2.2. System under Free-Motion	63

3. Parameter Sensitivity Study	65
4. Summary	67
CHAPTER 6. Robust Torque Control	68
1. System Linear Model and its Uncertainty	69
1.1. System under Constrained-Motion	70
1.2. System under Free-Motion	71
2. Control Objectives	73
3. Controller Design	73
4. Closed-loop Performance	76
4.1. System under Constrained-Motion	77
4.1.1. Frequency Domain Performance	77
4.1.2. Time Domain performance	78
4.2. System under Free-Motion	79
4.2.1. Frequency Domain Performance	79
4.2.2. Time Domain performance	80
5. Friction-Compensation	82
5.1. Friction-Compensation Algorithm	82
5.2. System Model and its Reduced Uncertainty	84
5.3. Robust Torque Control	85
5.4. Closed-Loop Performance Comparison	85
6. Torque Ripple Cancellation	86
6.1. Vibration Isolation	88
6.2. Vibration Absorbers	90
7. Summary	93
CHAPTER 7. Conclusions	94
1. Synopsis	94
1.1. Torque Sensing	94
1.2. Modelling and Simulation	95
1.3. Robust Torque Control	96
2. Directions For Future Research	96
REFERENCES	98
APPENDIX A. Mathematical Background	108

1. Least-Squares Estimation	108
1.1. Linear Regression Models	108
1.2. Nonlinear Regression Models	109
2. $\mathcal{H}_\infty$ Controller Design	110
2.1. Norms for Signals and Systems	110
2.1.1. The Signal Spaces $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{H}_2$	110
2.1.2. The System Spaces $\mathcal{L}_2$, $\mathcal{L}_\infty$ and $\mathcal{H}_\infty$	111
2.2. Feedback Control Systems: Expressions and Notations	112
2.2.1. Sensitivity and Complementary Sensitivity	113
2.2.2. Internal Stability	114
2.2.3. Performance Limitations	115
2.3. Uncertainty Representations	116
2.3.1. Multiplicative Uncertainty	116
2.4. Robust Stability	117
2.5. Robust Performance	119
2.6. Mixed-Sensitivity Problem	119
APPENDIX B. Software Code Listings	122
1. Open-Loop Time Response	122
1.1. The Host Portion	122
1.2. The C30 Portion	127
2. Torque Ripple Estimation by Kalman Filter	129
3. Torque Ripple and Misalignment Torque Estimation	136
4. Closed-Loop Time Response	140
4.1. Kalman Filtered Torque, $\mathcal{H}_\infty$ Controller and Friction Compensator	140
5. Open-Loop Frequency Response	146
5.1. The Host Portion	146
5.2. The C30 Portion	149
6. Closed-Loop Frequency Response	152
APPENDIX C. Modelling Scheme Verification Plots	157
1. System under Constrained-Motion	157
2. System under Free-Motion	161

LIST OF FIGURES

1.1	The components of harmonic drive transmission	2
2.1	A picture of the McGill experimental setup and its schematics for constrained-motion experiments	12
2.2	A picture of the McGill experimental setup for free-motion experiments	12
2.3	Speed-torque characteristics of the DC motor used in McGill setup .	13
2.4	A function diagram of the servo amplifier used in the McGill setup .	13
2.5	Efficiency of the harmonic drive used in the McGill setup working at different speeds and lubricated with HC-1 grease, taken from [63] . .	14
2.6	Pulse train from the encoder and the decoder	14
2.7	A picture of the ISE experimental setup	16
2.8	Speed-torque characteristics of the DC motor used in the ISE setup under continuous operation condition	16
2.9	A function diagram of the servo amplifier used in the ISE setup configured for velocity mode	17
2.10	A function diagram of the servo amplifier used in the ISE setup configured for torque mode	17
2.11	Major hardware subsystems of the Siglab	19
2.12	The schematics of the data acquisition and the processing hardware used for time response experiments	22
2.13	The schematics of the data acquisition and the processing hardware used for frequency response experiments	24
2.14	The flowchart of the typical programs executed on the host and on the C30 CPU for open-loop time response experiments	26

2.15	The flowchart of the typical programs executed on the host and on the C30 CPU for closed-loop time response experiments	28
2.16	The flowchart of the typical programs executed on the host and on the C30 CPU for closed-loop frequency response experiments	30
3.1	Position of four Rosette strain-gauges on the flexspline diaphragm . .	33
3.2	Strain-gauges Wheatstone bridge network	34
3.3	Proposed transparent film for accurate strain-gauge placement	35
3.4	Harmonic drive built-in torque sensor	35
3.5	Torque sensor calibration results for full-range loading	36
3.6	Measured Torque and velocity and the power spectrum of the measured torque with peaks at multiples of the velocity	37
3.7	Kalman filter performance to cancel torque ripples. Dotted: Measured torque, Solid: Kalman filtered torque	40
3.8	Misalignment torque signature on the measured torque	41
3.9	Kalman filter performance to cancel torque ripples, and misalignment torque for two typical experiments	42
4.1	Velocity comparison of the model and experiment for two typical experiments; Solid: simulations, dotted: measured velocity	47
4.2	Frequency response of the servo amplifier	48
4.3	Harmonic drive system modelled as a black box	49
4.4	Transmission model of the harmonic drive with compliance and friction	50
4.5	Typical hysteresis curve and its model for optimal α (dash-dot), and $\alpha = \frac{1}{2}$ (dashed)	51
4.6	Friction torque modelled as a superposition of Coulomb, viscous and Stribeck friction, having velocity-direction-dependent coefficient . . .	53
4.7	Comparison of output torque in simulations and experiments; dotted: experiment, solid: simulation	54
4.8	Identified friction curve	55

5.1	Harmonic drive system under constrained-motion simulated on Simulink for model verification purpose. Top: System; Mid: DC-motor; Bot: Harmonic drive	58
5.2	The modified friction function used for simulations to avoid chattering.	59
5.3	Harmonic drive system under free motion simulated on Simulink for model verification purpose. Top: System; Mid: DC-motor; Bot: Harmonic drive	60
5.4	Simulation verification for system under free and constrained-motion: Solid : Experiment, Dotted : Simulation	61
5.5	Harmonic drive system under constrained-motion simulated on Simulink	62
5.6	A typical simulation result of the harmonic drive system under constrained-motion	63
5.7	Harmonic drive system under free-motion simulated on Simulink . .	64
5.8	A typical simulation result of the harmonic drive system under free-motion for sinusoid input. Top: Torque, Mid: Wave generator velocity, Bot: Flexspline velocity	64
5.9	Comparison of the experiment with the complete and simplified models, where the significance of Coulomb and viscous friction are examined separately; Solid : Experiment, Dash-dotted : Complete model, Dashed : Simplified model	65
5.10	Comparison of the experiment with the complete and simplified models, where the significance of stiffness and structural damping are examined separately; Solid : Experiment, Dash-dotted : Complete model, Dashed : Simplified model	66
6.1	Frequency response of the system under constrained-motion, theoretical and nominal models, and multiplicative uncertainty	71
6.2	Frequency response of the free-motion system, nominal model, and multiplicative uncertainty	72
6.3	Block diagram of the system with multiplicative uncertainty for plant	73
6.4	Block diagram of the reconfigured system of Figure 6.3 suited for an $\mathcal{H}_\infty$ -framework	74

6.5	The frequency response of the two designed controllers; Solid : Constrained-motion, Dashed : Free-motion	75
6.6	Closed-loop frequency performance of the system under constrained-motion with weighting envelopes	76
6.7	Nyquist plot of the system under constrained-motion obtained from empirical frequency response of the closed loop system	77
6.8	Closed-loop time response of the system under constrained-motion .	78
6.9	Closed-loop frequency performance of system under free-motion . . .	79
6.10	Nyquist plot of the system under free-motion obtained from empirical frequency response of the closed loop system	80
6.11	Closed-loop time response of the system under free-motion	81
6.12	Identified Coulomb and viscous friction curve for harmonic drive systems	82
6.13	Block diagram of the friction-compensation algorithm	83
6.14	Frequency response of the free-motion system with friction compensation, its nominal model, and multiplicative uncertainty	84
6.15	Closed-loop frequency performance comparison of the system with and without friction-compensation; Dashed : With friction-compensation, Dash-Dotted : Without friction-compensation	86
6.16	Closed-loop time response comparison of the system with and without friction-compensation; Solid : Reference command, Dash-Dotted : With friction-compensation, Dotted : Without friction-compensation . . .	87
6.17	The model of vibration isolator as a spring and damper	89
6.18	Plot of the transmissibility ratio in logarithmic scale for a vibration isolator including a variety of damping ratios ζ	90
6.19	The model of a vibration absorber as aa inertia and a spring	91
6.20	Plot of the transmissibility ratio in logarithmic scale for vibration absorber with (dashed) and without damping (solid)	92
A.1	General representation of a system and its input and output	111
A.2	Block diagram for feedback control system	113
A.3	Block diagram for feedback control system with measurement noise .	115

A.4	Block diagram for multiplicative uncertainty representation	117
A.5	Block diagram for feedback control system with multiplicative uncertainty for plant	117
A.6	Feedback system used in small-gain theorem	118
A.7	Modified block diagram of system in Figure A.5 for robust stability analysis	118
A.8	The basic design configuration for the generalized plant G	120
C.1	Simulation verification: Sinusoid input, 80% of maximum amplitude.	157
C.2	Simulation verification: Asymmetric sinusoid input, with maximum amplitude.	158
C.3	Simulation verification: Asymmetric sinusoid input, 50% of maximum amplitude.	158
C.4	Simulation verification: Sinusoid input, 50% of maximum amplitude.	159
C.5	Simulation verification: Inverted sinusoid input, 50% of maximum amplitude.	159
C.6	Simulation verification: Triangular input, 50% of maximum amplitude.	160
C.7	Simulation verification: Composite input, 50% of maximum amplitude.	160
C.8	Simulation verification: 1.2 Hz sinusoid input	161
C.9	Simulation verification: 1.4 Hz sinusoid input	161
C.10	Simulation verification: 1.6 Hz sinusoid input	162
C.11	Simulation verification: 1.8 Hz sinusoid input	162
C.12	Simulation verification: 2 Hz sinusoid input	163
C.13	Simulation verification: 2.5 Hz sinusoid input	163

LIST OF TABLES

1.1	Performance advantages of harmonic drive systems	3
1.2	Performance disadvantages of harmonic drive systems	4
2.1	Technical specification of the DC motors used in the McGill and the ISE experimental setups	15
2.2	Technical specification of the harmonic drives used in the McGill and the ISE experimental setups	15
3.1	Torque sensor calibration results	36
4.1	DC motors estimated parameters	46
4.2	Harmonic drives estimated parameters	52

CHAPTER 1

Introduction

Robotic manipulators require actuators with high-torque capability at low-velocities. Commercial DC-motors however, provide their operating torque only at high-velocities. Many electrically actuated robots therefore, use a gear transmission to increase the torque and decrease the drive speed. Among gear transmissions, harmonic drives has captured the attention of designers, because they are high-ratio, compact and light-weight mechanisms with almost no backlash. Harmonic drive transmission employs a non-rigid gear called *flexspline* for speed reduction. As a consequence, the transmission stiffness is lower than that in conventional transmissions. Furthermore, the nonlinear relation between the input and the output torques makes it more challenging to control the system. Following this impetus for investigation, several researchers have made valuable contributions to the technological status and control of harmonic drive systems. In this thesis the robust torque control of harmonic drive systems is examined in details. The research objectives pursued in this research is elaborated in this chapter, in addition to introducing harmonic drive systems and the research literature surrounding harmonic drives.

1. Harmonic Drive Systems

Developed in 1955 primarily for aerospace applications, harmonic drives are high-ratio and compact torque transmission systems that have enjoyed wide industrial applications. This mechanical transmission, occasionally called “strain-wave gearing”, employs a continuous deflection wave along a non-rigid gear to allow for gradual engagement of gear teeth. Because of this unconventional gear-tooth meshing action, harmonic drives can deliver high reduction ratios in a very small package.

1.1. Principles of Operation

Every harmonic drive consists of the three components as illustrated in Figure 1.1. The wave generator is a ball bearing assembly with a rigid, elliptical inner race and a flexible outer race. The flexspline is a thin-walled, flexible cup adorned with small, external gear teeth around its rim. The circular spline is a rigid ring with internal teeth machined along a slightly larger pitch diameter than those of the flexspline. When assembled, the wave generator is nested inside the flexspline, causing the flexible circumference to adopt the elliptical profile of the wave generator, and the external teeth of the flexspline to mesh with the internal teeth on the circular spline along the major axis of the wave generator ellipse.

If properly assembled, all three components of the transmission can rotate at different but coupled velocities on the same axis. To use the harmonic drive for speed reduction, the wave generator is mounted on the electric motor shaft, and the output is conveyed either through the flexspline while the circular spline is fixed or through the flexspline while the circular spline is fixed. In the latter case, by rotation of the wave generator the zone of gear-tooth engagement is carried with the wave generator major elliptical axis. When this engagement zone is propagated 360° around the circumference of the circular spline, the flexspline, which contains fewer teeth than the circular spline, will lag by that deficit number of teeth relative to the circular spline. Through this gradual and continuous engagement of slightly offset teeth, every rotation of the wave generator moves the flexspline a small angle back on the circular spline. By this unconventional mechanism, gear ratios up to 320 : 1 can be achieved in a single transmission.

Since the harmonic drive has three rotational ports, by using different combinations of rotations on these ports, numerous differential-gearing functions and reduction ratios can

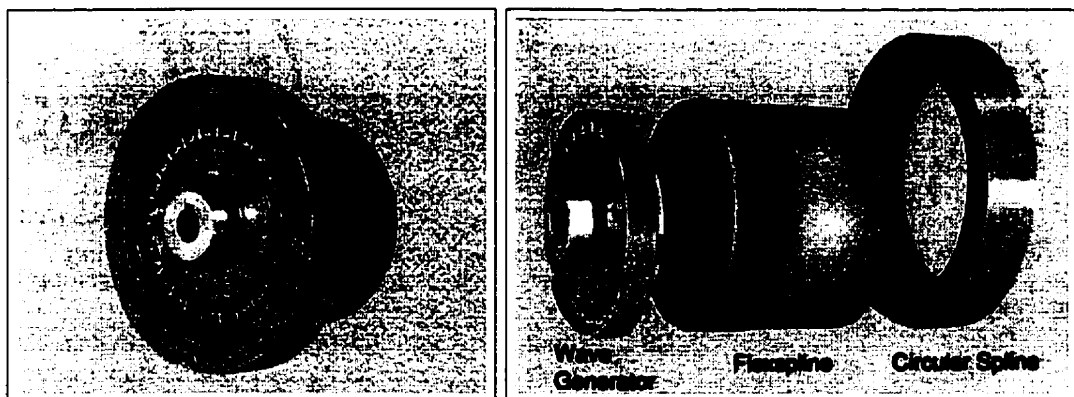


FIGURE 1.1. The components of harmonic drive transmission

be achieved. In its most popular configuration, the circular spline is fixed to the ground and a low-torque, high-speed motor drives the wave generator to produce a high-torque, low-speed rotation on the flexspline.

1.2. Performance Features

The harmonic drive displays performance features both superior and inferior to conventional gear transmissions. These performance features are categorized in Tables 1.1 and 1.2. Typical gear ratios of commercial drives range from 50:1 to 320:1, with efficiencies approach 90%. The smallest transmission can provide respectable maximum torque output of about 3.5 N-m, while heavy-duty units boast up to 10,000 N.m of torque capacity.

The unique performance features of the harmonic drive has captured the attention of designers in many fields. It has been used in industrial robots, assembly equipment and measuring instruments, as well as heavy duty applications such as machine tools and printing presses. Additionally, space and aircraft systems often employ harmonic drives for their light weight and compact geometry.

TABLE 1.1. Performance advantages of harmonic drive systems

High torque capacity	Since torque is transmitted through multiple-teeth, harmonic drives can withstand high torque at small pitch diameters.
The concentric geometry	Since all three harmonic drive components are concentric and coaxial, designers can drastically reduce power-train size and complexity.
Lightweight and compact	Requiring only three basic elements, the harmonic drive can deliver extremely high gear ratios in a small package.
Zero backlash	Natural gear preloading and predominantly radial tooth engagement virtually eliminate all transmission backlash.
High efficiency	If properly lubricated, typical efficiencies of harmonic drive transmission can reach 80 to 90 percent.
Back drivability	Due to their high efficiency, wave generator rotation can be driven through the flexspline or circular spline.

TABLE 1.2. Performance disadvantages of harmonic drive systems

High flexibility	Due to the high flexibility of the flexspline, moderate operating torques can produce substantial transmission torsion.
Resonance vibration	Since torque ripples produced by gear meshing can interact with the low stiffness of the transmission to excite resonances, high vibration amplitudes is generated in some operating ranges.
Nonlinearity	Both compliance and frictional losses in the drive constitute nonlinear behaviour.
Low insight knowledge	Compared to the conventional gear transmissions, relatively little exploration has been done of the unusual operating mechanisms in the harmonic drive.

2. Research Objectives

In numerous robotic control techniques such as feedback linearization, computed torque method and some adaptive control schemes, the actuator torque is taken to be the control input [89, 99, 100]. The physical variable being manipulated in practice, however, is not torque but armature current in a DC motor, for instance. For harmonic drive systems the relation between output torque and input current possesses nonlinear dynamics, due to the flexibility, Coulomb friction and structural damping of the harmonic drive [103]. The objective of this research is *to improve this input/output relation by torque feedback and to convert the system to a near-ideal torque source with a flat frequency response over a wide bandwidth.*

The first step to accomplish this objective is to characterize the nonlinear dynamics of the harmonic drive system as a mathematical model. The goal of modelling the harmonic drive system is to discover a simple representation which can predict system behaviour to a desired level of accuracy. Our particular goal in modelling is to ultimately implement a *model-based* torque control algorithm on the system. Moreover, we used the computer model for examining and improving control laws, prior to implementing them. As recommended by other researchers [76, 121], in order to achieve this objective, it is necessary to have at least a model for friction and compliance of harmonic drive systems. One of the contributions of this thesis is the development of simple and accurate representations for the friction and compliance of the harmonic drive systems. In practice the process of modelling and

identification not only provides better understanding of the system dynamics [106], but also leads to new horizons for model-based torque controller design [104].

In order to apply torque feedback on the robot joint, it is necessary to measure the transmitted torque through the actuator transmission mechanism. Conventionally, torque sensor are placed in the output transmission line of the robot [58, 101]. However, for a harmonic drive transmission, which has an elastic element, *i.e.* flexspline, there is no advantage in adding an additional compliant element and thereby reducing the joint stiffness. An *intelligent built-in torque sensor* is developed in this research in order to measure the transmitted torque. As proposed by Hashimoto in 1989 [49], strain-gauges are directly mounted on the flexspline to measure the torque. The torque sensing technique is further developed by filtering the torque ripples via kalman filter estimation. This method proved to be an economical and effective way of torque sensing for harmonic drives in our setup [109, 110].

The torque-control of harmonic drive system is addressed next through this research. There is a dichotomy of torque-control applications for a robotic manipulator using harmonic drives in its joints. First are applications where the robot is in contact with a stiff environment, and high torques at very low velocities are required at each joint. Simulation of this application at each joint can be studied by a constrained-motion experiment. The second class of applications occur when the robot arms are moving freely, and the torque required at each joint is to compensate for gravity, friction and link acceleration only. This problem can be simulated through a free-motion case, especially where the gear ratio is large enough for the motor inertia to dominate. In the free-motion case, the amount of torque required at each joint is very low but at much higher velocities.

The main contribution of this research is in the design and implementation of torque-controllers for harmonic drive system for both constrained-motion [107] and free-motion applications [105]. The controllers are designed using frequency domain representations of the system, and are implemented using Kalman filtered torque as feedback. From ensembles of empirical frequency response estimates, the nonlinear system is encapsulated by a linear model and uncertainty representation. Robust torque controllers are designed using this information in an $\mathcal{H}_\infty$ -framework. Exceptional performance results are obtained from the time and frequency response of the closed-loop system, especially for constrained-motion application. To further improve the performance of the system for free-motion application, a model-based friction-compensation controller is implemented in addition to the $\mathcal{H}_\infty$ torque control. It is shown that friction-compensation shrinks the model uncertainty

at low frequencies and hence, the performance of the closed-loop system is improved at those frequencies [104].

3. Literature Survey

Throughout its short existence, the harmonic drive has enjoyed increasing international attention from designers as well as researchers. This section provides a brief survey of the literature on harmonic drive research according to four topics: technological status, torque sensing, modelling and identification, and control of harmonic drive systems.

3.1. The Technological Status

Manufacturer's catalogues provide a wide range of information on the technological status of harmonic drive systems [60–64, 111–113]. Charlson outlines the potential of harmonic drives in servomechanisms, and provides a technical review of the harmonic drive performance [22], while Kiyosawa and Sasahara reported the performance characteristics of the harmonic drive system using a new tooth profile [77]. Thompson [115], and Hui et al. [58] are among the researchers who designed robotic manipulators actuated with harmonic drive systems for further research and development. Schempt and Yoerger performed a comparison study of the harmonic drive and the other transmissions [96]. A laboratory-scale design of flexspline from composite materials is recently reported in the literature [68]. The last thirty years of technological development of harmonic drives provided designers in disparate fields with an increasing potential in using harmonic drive system for a wider range of applications, especially in aerospace and robotics.

3.2. Torque Sensing

In order to apply torque feedback on a robot joint, it is necessary to measure the transmitted torque through the actuator transmission mechanism. Conventionally, torque sensor are placed in the output transmission line of the robot [54, 58, 101]. However, for a harmonic drive transmission, which has an elastic element, the flexspline, there is no advantage in adding an additional compliant element and thereby reducing the joint stiffness. In the Chapter 3 we use and further develop the idea of a built-in torque sensor for harmonic drives following Hashimoto in 1989 [49]. By an analytic/experimental study, Hashimoto et al. illustrated the performance of using built-in torque sensors for harmonic drives [49, 51], and further used the built-in torque sensor information for torque feedback [50, 52, 53]. In our study, we used the idea of built-in torque sensor, and compensated the measured torque signal from torque ripples and misalignment torque signatures [109, 110]. As explained

in Chapter 3, torque ripples are a high frequency oscillation in the output torque signal, and has been observed by other researchers [24, 42, 124]. Godler et al. and Chiba proposed control schemes to reduce the effect of torque ripples in the output velocity of the system [24, 42].

3.3. Modelling and Identification

The Russians were perhaps the first to initiate substantial research on the dynamic behavior of harmonic drives [2, 67, 98, 123]. More recently Tuttle and Seering devoted an extensive effort to model the stiffness, positioning accuracy, gear tooth-meshing mechanism and friction of harmonic drives [118–121]. Their experimental observations show that the velocity response to step commands in motor current are not only contaminated by serious vibration, but also by unpredictable jumps. The velocity response observations were used to guide the development of a series of models with increasing complexity for harmonic drive behavior. Their most complex model involve kinematic error, nonlinear stiffness, and gear-tooth interface with frictional losses.

Kircanski and Goldenberg have also attempted to model the harmonic drive in detail [76]. They used the drive system in contact with a stiff environment, as opposed to the free-motion experiments used by Tuttle and Seering [121]. They illustrated that for their case, nonlinear stiffness, hysteresis and friction are more tractable. Simple models for soft-windup, hysteresis and friction were proposed and the parameters were identified by restrained motion experiments.

Hsia [57], Legnani [79], Marilier [83], Chedmail et al. [23] and Seyfferth [97] are among others who attempted to model the stiffness, friction and position accuracy of harmonic drive systems. All these researchers noted the inherent difficulties in finding an accurate model for the system.

In our research simple and accurate models for compliance, hysteresis and friction are proposed. There is a rich literature available for the modelling, identification and compensation of friction. The history of friction models dates back to (1519) to Leonardo da Vinci [28]. Tustin [117], Dahl [29] and Threlfall [116] are among the contributors in experimental verification of friction models. These models are further modified and justified for different applications [15, 55, 71]. Armstrong devoted his research on experimental determination, modelling and compensation of friction. [3–9]. Reference [9] provides an important survey of modelling, identification and compensation schemes for friction. Dupont [38, 39],

Dohring [33], Gogoussis and Donath [43], Terauchi [114] and Gomes [45] are the representative work on friction models used in robotic transmissions. Special attention is given to computer simulation of friction models, because of its inherent hard nonlinearity. Bernard [14], Karnopp [70] and Haessig and Friedland [48], are among the researchers who addressed friction simulation.

Many researchers have attempted to model harmonic drive compliance. Volkov and Zubkov [124], Aliev [2], Margulis [82] and Legnani [79], are among those researchers who provide different models for harmonic drive stiffness. More recently Kircanski and Goldenberg [76] and Tuttle and Seering [121] provide experimental verification of stiffness models for harmonic drives. The hysteresis effect however, is more difficult to model, and consequently is often ignored. Recently Seyfferth et al. proposed a fairly complex model to capture the hysteresis figure in harmonic drive torsion curves [97]. Other mathematical models for hysteresis are available in the literature, from which Mayergoyz et al. [84,85], Ossart and Meunier [91] and Macki et al. [81] provide comparison studies among the models.

3.4. Control

Brigdes et al. [18], Kaneko et al. [69], Kazerooni [72,73], Hogan [56], Chapel and Su [21] and Kubo et al. [78] are representative of researchers who worked on the control of harmonic drive system. Bridges used a very simple linear model for the system, with PD torque control [18]. His results show some improvement in tracking error, but insufficient performance near resonant frequency. Kaneko also based his analysis on a simple model of the system, but included nonlinear stiffness in the system [69]. He then applied a feedforward loop to adjust for nonlinear stiffness and then a pure gain torque feedback to shape the performance. Kazerooni considers a simple linear system for the harmonic drive, and used a sensitivity loopshaping technique to design a linear controller for the system [73]. Hogan proposed impedance control for robots with harmonic drive systems, to deal with the dynamic interaction introduced in contact tasks [56], while Chapel applied $\mathcal{H}_\infty$ control design methods to the analysis and design of impedance control laws [21].

Kubo et al. examined friction-compensation on harmonic drives, and presented a stability analysis with experimental verification of the improved performance of the system [78]. Friction compensation for robotic manipulators is addressed by other researchers. Canudas de Wit [30–32], and Walrath [125] have tested adaptive friction compensation schemes for robotic manipulators. Adams and Payandeh provides methods for low-velocity friction compensation [1], and Cai and Song successfully compensated for stick slip friction

[19]. Finally, Bona and Indri [16], and De Carli et al. [20] are among the researchers who have used friction compensation techniques in practice.

4. Thesis Outline

This thesis is composed of seven chapters. The harmonic drive experimental testing stations are described in Chapter 2. Two harmonic drive testing stations are used in this research to respectively monitor the behaviour of two different harmonic drives. The hardware and software used to perform open- and closed-loop experiments are also introduced in this chapter. Chapter 3 is devoted to the design and implementation of an intelligent built-in torque sensor for harmonic drives. In this sensor four strain-gauges are directly mounted on the flexspline by an accurate positioning method, in order to minimize the sensing inaccuracy. Furthermore, to cancel torque ripples, the oscillation observed on the measured torque, Kalman filter estimation is employed. A simple fourth-order harmonic oscillator proved to accurately model the torque ripples. Moreover, the error model is extended to incorporate any misalignment torque. By on line implementation of the Kalman filter, it is shown that this method is a fast and accurate way to filter torque ripples and misalignment torque. Hence, the intelligent built-in torque sensor introduced in Chapter 3 is a viable and economical way to measure the harmonic drive transmitted torque, for the purpose of torque feedback.

In Chapter 4 a methodology to model harmonic drive systems and to identify model parameters is presented. Simple and accurate models for compliance, hysteresis and friction are proposed, and the model parameters are estimated using least-squares approximation. A statistical measure of variation is defined, by which the reliability of the estimated parameter for different operating condition, as well as the accuracy and integrity of the proposed model is quantified. In Chapter 5 different simulation for the system under constrained- and free-motion are developed. First the simulations used to verify the model by experiments are presented. Then the general simulation of the system for constrained- and free-motion applications are given, and finally, the significance of the nonlinear model is assessed through a parameter sensitivity study.

Chapter 6 is devoted to the design and implementation of robust torque controller for harmonic drive systems in an $\mathcal{H}_\infty$ -framework. Torque control of system for constrained-motion and free-motion applications is examined in detail. A Nominal model for the system is obtained from experimental frequency response estimates, and the deviation of the system

from the model is encapsulated by a multiplicative uncertainty. Robust torque controllers are designed using this information in an $\mathcal{H}_\infty$ -framework, and implemented on two different setups. Appendix A introduces the mathematical background and the terminology which has been used in this chapter for $\mathcal{H}_\infty$ controller design. From time and frequency domain experiments, it is shown that the closed-loop system maintains robust stability and improves the tracking performance exceptionally well. To further improve the performance of the system for free-motion application, a model-based friction-compensation algorithm is implemented in addition to the robust torque control. It is shown that friction-compensation shrinks the model uncertainty at low frequencies and hence, the performance of the closed-loop system will be improved at those frequencies. Chapter 6 is concluded by presenting different methods to cancel torque ripples from the output torque. Finally, conclusions and future research directions are given in Chapter 7.

CHAPTER 2

The Experimental Setup Hardware & Software

1. Harmonic Drive Testing Stations

Two harmonic drive testing stations were used to respectively monitor the behaviour of two different harmonic drives. Several open-loop and closed-loop experiments are performed on both setups. Time and frequency response experiments are applied to the open-loop system for the purpose of modelling and parameter identification of the system, while closed-loop time and frequency response experiments are performed in order to analyse the controller performance. The first testing station was developed entirely at the Centre for intelligent machines, McGill university. The second setup was also designed and constructed at McGill, but the components were provided by International Submarine Engineering Ltd. In both setups the harmonic drive is driven by a DC motor, and a load inertia is used to simulate the robot arm for free-motion. Also a positive locking system is designed so that the output load can be locked to the ground for constrained-motion experiments. In the McGill experimental setup, the circular spline is fixed to the ground and the output is carried by the flexspline, while in the ISE setup, the flexspline is fixed and the circular spline is used for output rotation. By this arrangement, the behavior of the transmission under two different operating configurations is examined. Both setups are equipped with a tachometer to measure the motor velocity, and an encoder on the load side to measure the output position for free-motion experiments. The current applied to the DC motor is measured from the servo amplifier output and the output torque is measured by a Wheatstone bridge of strain gauges mounted directly on the flexspline. The details of torque sensing technique is elaborated in Chapter 3.

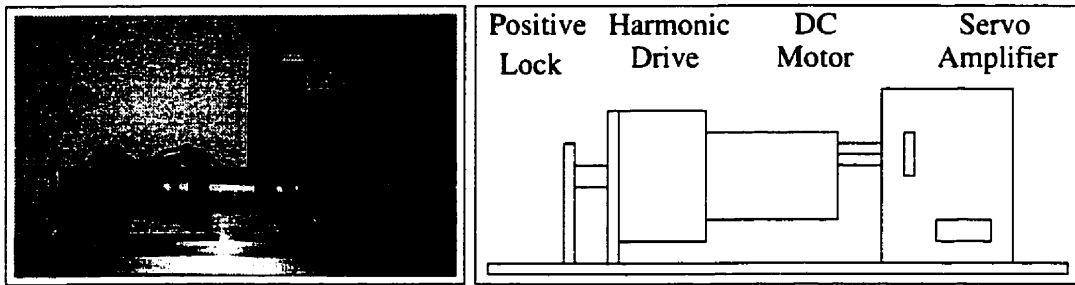


FIGURE 2.1. A picture of the McGill experimental setup and its schematics for constrained-motion experiments

1.1. McGill Setup

A picture of the McGill experimental setup and its schematics for constrained-motion experiments are illustrated in Figure 2.1, while Figure 2.2 illustrates the setup used for free motion experiments. In the McGill setup, a permanent magnet brushed DC motor from Electro-Craft, model 586-501-113, is used [25]. Its weight is 1.36 Kg , with maximum rated torque of 0.14 Nm , and its torque constant is 0.0543 Nm/amp . The technical specification of the DC motor is given in Table 2.1, and the speed-torque characteristics is given in Figure 2.3.

To drive the DC motor, a 100 Watt Electro-Craft servo amplifier, model Max-100-115, is used [25]. The servo amplifier can be configured to work in “velocity” or “torque mode”. When programmed in torque mode the amplifier interfaces with a controller, which provides an analog voltage signal proportional to the desired motor current. Ideally, the servo amplifier can be considered as a current source when configured in “torque mode”. The effect of back-EMF however, limits the performance of servo amplifier. Similarly, in “velocity mode” the amplifier interfaces a velocity feedback, which provides an analog voltage signal proportional to the desired motor velocity. Figure 2.4 provides a functional diagram for

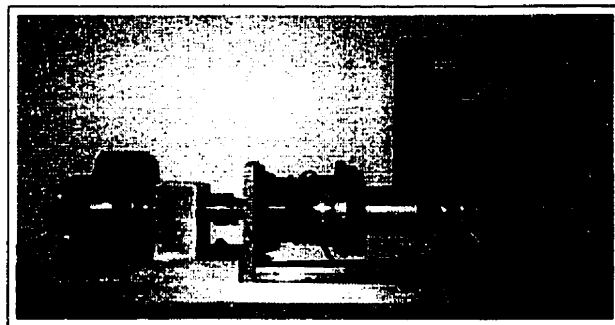


FIGURE 2.2. A picture of the McGill experimental setup for free-motion experiments

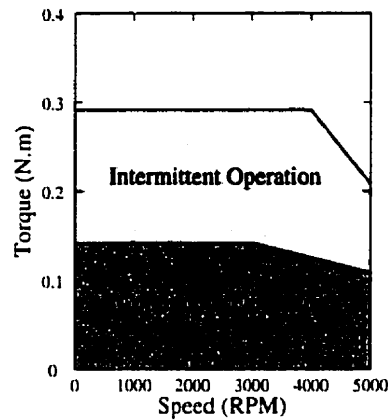


FIGURE 2.3. Speed-torque characteristics of the DC motor used in McGill setup

the current and velocity feedback used in the servo amplifier. The servo amplifier contains circuitry for protection of both motor and amplifier from effects of various overload and short circuit conditions. The servo amplifier is a 25 kHz pulse-width-modulation (PWM) switching type that minimizes system power losses. It also contains automatic current fold-back if the RMS current exceeds ± 3 Amps, while maximum ± 6 Amps is allowed for 50 ms duration. This feature protects the motor and amplifier from over-current conditions. The motor supply voltage for this type of amplifier is 36 Volts DC, with continuous power rating of 100 Watts.

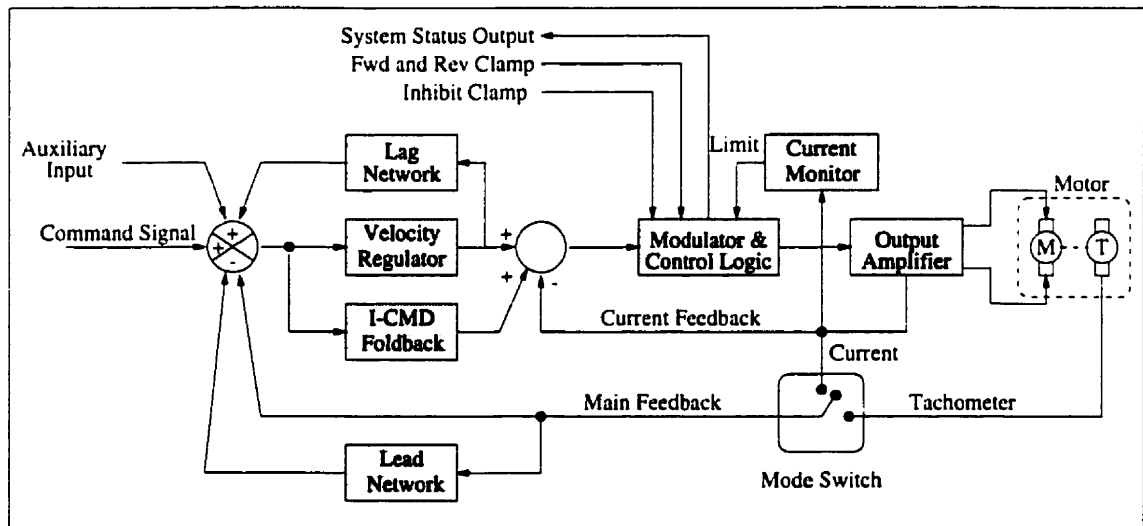


FIGURE 2.4. A function diagram of the servo amplifier used in the McGill setup

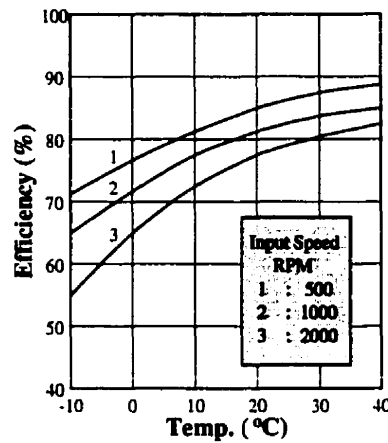


FIGURE 2.5. Efficiency of the harmonic drive used in the McGill setup working at different speeds and lubricated with HC-1 grease, taken from [63]

The harmonic drive in this setup is from the RHS series of HD systems, model RHS-20-100-CC-SP, with gear ratio of 100:1, and rated torque of 40 Nm [63]. Other technical specification of the harmonic drive is given in Table 2.2. The efficiency of this type of harmonic drive is a function of operating temperature, speed, and lubrication conditions. Figure 2.5 illustrates the efficiency of the harmonic drive at different speeds and lubricated with harmonic drive grease HC-1 [63], in which 80 – 90 % efficiency is achievable at normal operation conditions.

An incremental-type shaft encoder is used in the setup to measure the output rotation. The encoder is an “Optical Hollow Shaft Encoder” From RENCO, model RM21, with a resolution of 1000 pulses/rev. It has two pulse train channels A and B having a variable frequency that is linearly proportional to the shaft velocity. The phase between these two channels is $\pm 90^\circ$, where the polarity of the phase shift depends on the rotational

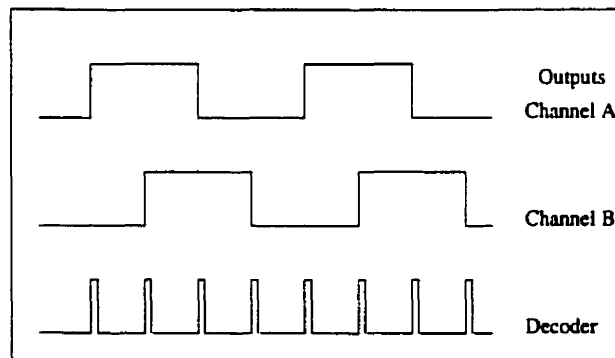


FIGURE 2.6. Pulse train from the encoder and the decoder

direction. As illustrated in Figure 2.6, the decoder, which converts the channel information into pulse count, generates pulses at the leading edge and trailing edge of the signals from both channels. Hence, the pulse counter picks up four times the pulses emitted by either channels of the encoder during one sampling period, and the overall resolution of the system increases to 4000 counts/rev.

TABLE 2.1. Technical specification of the DC motors used in the McGill and the ISE experimental setups

Motor Ratings	Units	McGill Motor	ISE Motor
Continuous Torque	N.m	0.14	0.63
Stall Torque	N.m	0.28	0.68
Max. Speed	RPM	5000	1230
Peak Current	Amp	6	12.6
Max. Continuous Current	Amp	3	10
Max. Continuous Power	Watts	100	109.5
Rotor Moment of Inertia	Kg.cm ²	0.368	0.44
Torque Constant	N.m/Amp	0.0543	0.182
Voltage Constant	V/KRPM	5.7	19.1
Resistance @ 25°C	Ω	1.2	1.9
Inductance	mH	1.92	1.6
Weight	Kg	1.36	0.482
Tach. Voltage Constant	V/rad/sec	0.137	0.033

TABLE 2.2. Technical specification of the harmonic drives used in the McGill and the ISE experimental setups

Ratings	Units	McGill HD	ISE HD
Gear Ratio	-	100 : 1	160 : 1
Rated Input Speed	RPM	2000	2000
Max. Input Speed	RPM	4000	4000
Rated Output Torque	N.m	40	67
Repeated Peak Torque	N.m	82	176
Momentary Peak Torque	N.m	148	314
Moment of Inertia	Kg.cm ²	0.819	0.428

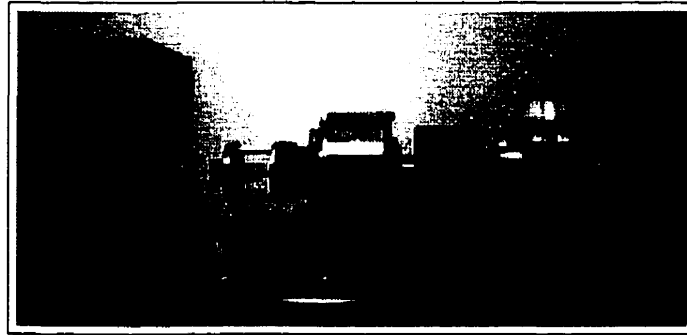


FIGURE 2.7. A picture of the ISE experimental setup

1.2. ISE Setup

A picture of the ISE setup is illustrated in Figure 2.7. The DC motor is a three-phase, brushless Kollmorgen Inland motor, model RBE-01503-A00 [87]. Its weight is 475 *gr*, with maximum rated torque of 5.6 *Nm*, and torque constant of 0.1815 *Nm/amp*. The technical specification of the DC motor is given in Table 2.1. The speed-torque characteristics of the DC motor is given in Figure 2.8. There is a built-in Hall sensor in the DC motor used for ISE setup, which provides position and current measurements.

To drive the DC motor a 100 watts FAST Drive Kollmorgen servo amplifier, model FD 100/ 5E1 is used [88]. A FAST (Flexible Amplifier Servo Technology) Drive is a fully digital, PWM servo amplifier, which can power a broad range of motors including brush and brushless permanent magnet DC motors. Through the use of digital signal processors (DSP) and a field programmable gate array (FPGA), the FAST drive is a flexible servo, virtually eliminating changes in hardware for various amplifier configurations. The motor type, pole count, feedback type, and motor phasing are automatically configured, and advanced

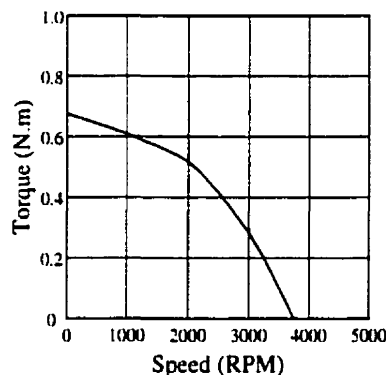


FIGURE 2.8. Speed-torque characteristics of the DC motor used in the ISE setup under continuous operation condition

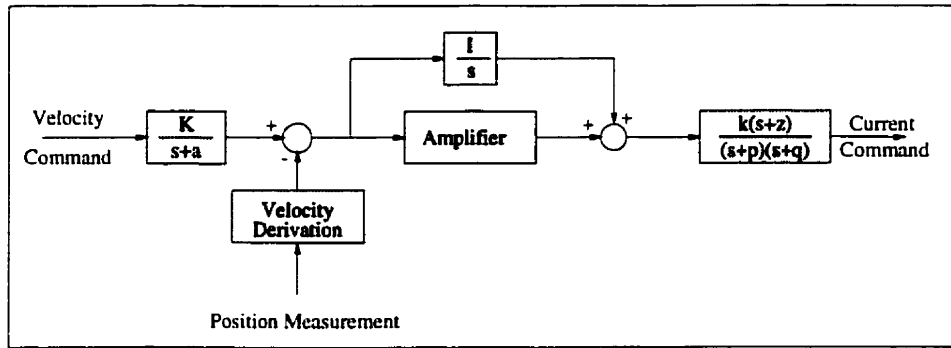


FIGURE 2.9. A function diagram of the servo amplifier used in the ISE setup configured for velocity mode

features such as loop type, loop compensation, programmable current limit, phase advance, and communication mode are user configurable via the RS-232 serial port. Similar to the servo amplifier used in McGill setup, the FAST drive can be configured to work in velocity or torque mode. Figures 2.9 and 2.10 illustrated the feedback functional diagram of the servo amplifier under velocity and torque modes, respectively. For torque feedback applications, the servo amplifier is configured into torque mode. The structure of the velocity feedback in the servo amplifier used in ISE setup is similar to that of McGill setup. For current mode however, the feedback system in the FAST drive accommodates the three-phase structure of the DC-motor. The internal feedback element of the FAST drives are implemented using DSP control, hence, the configuration of the feedback system can be flexibly adjusted [88].

The harmonic drive used in the ISE setup is from the CFS series of HD Systems, Inc. with gear ratio 160:1, and rated peak torque of 178 Nm [61]. Other technical specification of the harmonic drive are given in Table 2.2. The major difference of CFS series to RHS series used in the McGill setup is that the length of the flexspline, and therefore the overall

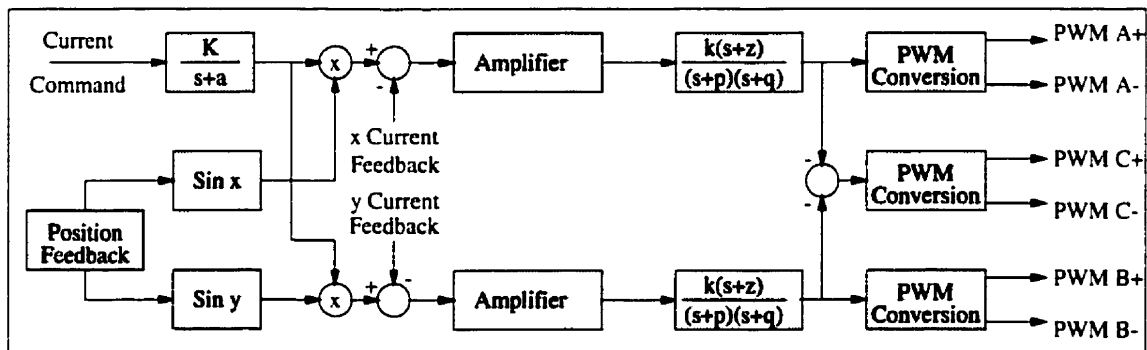


FIGURE 2.10. A function diagram of the servo amplifier used in the ISE setup configured for torque mode

length of the harmonic drive, is smaller in CSF series. This design modification increases the stiffness of the harmonic drives. The encoder used in the ISE setup is the same as that used in the McGill setup.

2. The Siglab Hardware

The DSP Technology's Siglab hardware is used for frequency response experiments for the harmonic drive testing stations [47]. The Siglab is a PC-based DSP hardware, which requires Matlab V4.2 for numerical analysis of the input-output measurements. It features the graphical displays of Matlab graphical user interface (GUI) in the windows environment. As illustrated in Figure 2.11, the Siglab has three major hardware subsystems: control processor and input and output subsystems. The control processor consists of a TMS320C31 DSP from Texas Instruments, with 256 Kbytes of SRAM for programming, 1–8 Mbytes of DROM for data handling, SCSI controller unit and serial I/O buffers. The input subsystem includes two ports, each consisting of a $\Sigma \Delta$ A/D converter, analog anti-alias filter, DSP filter, and variable gain and offset. It also includes circuitry for over-current protection and external signal triggering. The output subsystem also includes two channel DSP signal generators. Each channel consisting of a $\Sigma \Delta$ D/A converter, analog image filter and DSP filter. The two output ports of Siglab can be used to excite the system under experiment with a wide range of DSP generated signals, while the two input port can be used for spectrum analysis, or network identification. There are six virtual instruments (VIs) available in the Siglab, which allow measurements to be made on a dynamic system directly in the Matlab environment. We use specifically the virtual network analyzer (VNA) and the virtual sine-sweep (VSS) for the harmonic drive frequency response experiments.

2.1. Virtual Sine-Sweep (VSS)

The sine-sweep is one of the oldest network analysis technique to estimate a frequency response estimate of a dynamic system, and it is still the best when faced with systems with high measurement noise. In the absence of noise, FFT-based estimation techniques used by VNA is more convenient because of their speed. For a frequency response estimate of our harmonic drive systems, both methods were employed to get more reliable results.

In the sine-sweep technique, the system is excited using sine waves of increasing frequency through a desired frequency range. In Siglab, the reference channel (channel 1) measures the excitation signal while the response channel (channel 2) measures the output of the system under test. Both input channels are measured using a narrow-band tracking

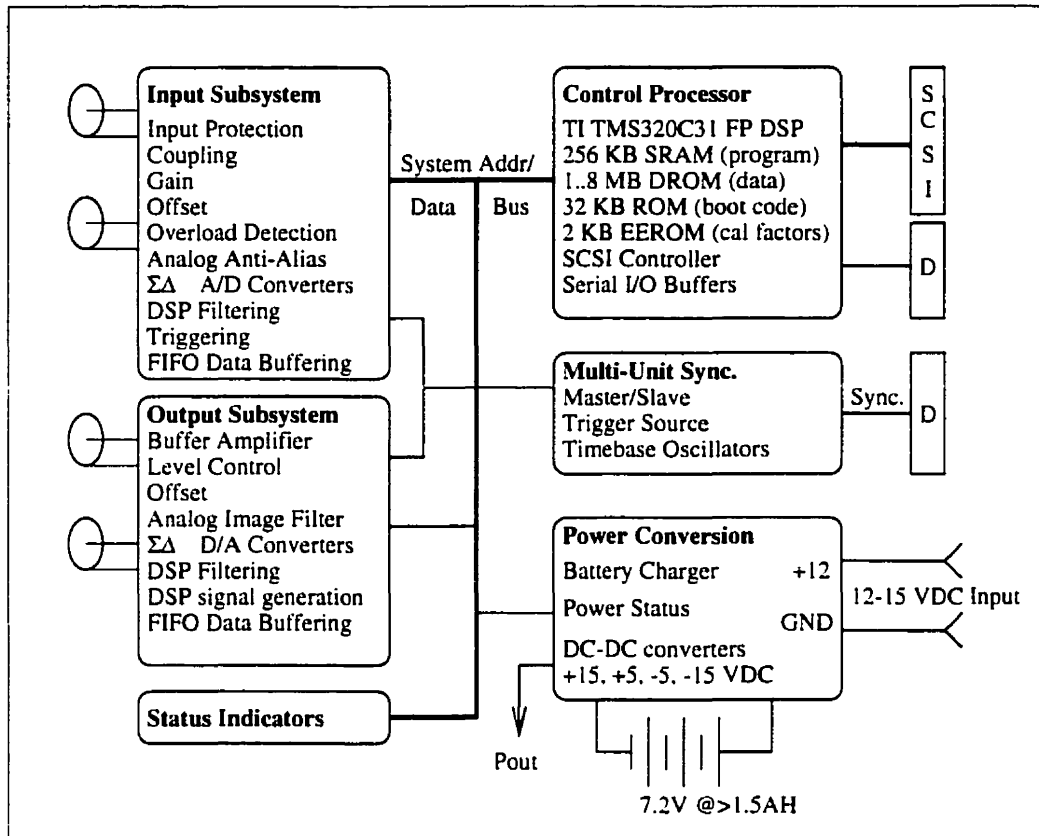


FIGURE 2.11. Major hardware subsystems of the Siglab

filter whose passband is centred on the excitation frequency at each step in order to reduce the effect of noise. The ratio of the sine wave amplitudes is displayed on-line as the frequency response estimate. The computation involves finding the ratio of the cross-spectrum of input/output and the input auto-spectrum which results in both the magnitude and the phase information of the frequency response estimate. A coherence function plot is also available at the display, which is valued between zero and one. A value of one indicates an ideal estimation, while the closer the value is to zero the poorer is the estimation. Therefore, the frequency response estimate is inaccurate at frequencies that have low coherence, so more experiments are required to average over an accurate estimate. The final frequency response estimate and the coherence function can be saved as Matlab data files for further analysis.

2.2. Virtual Network Analyzer (VNA)

The VNA is optimized to estimate the frequency response of a dynamic system using broad-band FFT techniques. In this method the system is excited by a chirp or random noise signal, which is measured by channel 1 input of the Siglab, while the output of the system is measured by channel 2. The bandwidth of the excitation signal is automatically adjusted from the user-defined frequency range of estimation. Similar to the VSS the computation of the frequency response estimate involves finding the ratio of cross-spectrum of the input/output and the input auto-spectrum. This estimation techniques works well if the response channel has a good signal to noise ratio. The chirp excitation is preferable to random, for nearly linear system. Random noise excitation however, makes a better estimation on systems with nonlinearity at the expense of longer measurement time and reduced frequency resolution [47].

The VNA provides online transfer function display during the averaging in different formats. Among them linear, logarithmic and dB scales of the magnitude plot can be displayed as well as phase plot in degrees. Similar to VSS, the coherence function is also calculated and displayed as the experiment evolves. The final frequency response estimate and the coherence function can be saved as Matlab data files for further analysis.

2.3. System Identification

From the frequency response estimates obtained from VNA or VSS, the system can be characterized by a transfer function. Siglab provides a virtual system identification module (VID), which fits the frequency response estimates with a causal transfer function. In our experiments however, we used the system identification toolbox of Matlab in order to find a transfer function estimate for the system. The system identification toolbox of Matlab incorporates more effective routines than VID, which enable us to not only find a nominal transfer function estimate for the system, but also to encapsulate the system variation by an uncertainty representation. The details of identification method used for harmonic drive system is given in § 1 of Chapter 6.

3. Data Acquisition and Processing Hardware

A schematic of the data acquisition and processing hardware used for time response experiments is illustrated in Figure 2.12. As explained in § 1, in the harmonic drive testing station velocity, current, torque and angular position is measured by various measurement units. Except for the encoder which provides a digital output, the other measurements

are analog. As illustrated in Figure 2.12, a 16-channel D/A converter (DT 1401) is used for digital conversion of these analog signals, while an encoder interface (Proto XVME-085) provides the encoder readings for further processing. The central processing unit (CPU) is a Challenger-C30V multi-processor, which is based on two TMS320C30 DSP chips from Texas Instruments. All the calibration and signal processing is done on-line by this CPU. An eight channel D/A converter (VMEVMI 4116) is used to command the reference signal generated in the CPU for the servo amplifier. All the data acquisition and processing hardware are assembled in a 19-inch Versa Modula Europa (VME) chassis, which in turn is connected to the host computer (Sun Sparc Station). An S-bus to VME-bus adapter board housed in the chassis maps the VME addresses into the memory of the Sun Sparc Station. The computational engine for real-time implementation of signal generation, measurement processing, and control is the dual TMS320C30 DSP, while the host computer is for cross-compiler software development, VME interfacing, and off-line post-processing of the experimental data.

3.1. The Challenger-C30V Specifications

The Challenger-C30V is a high-speed, system oriented, digital signal multi-processor [65]. The Challenger provides a floating-point performance level of 66 MFLOPS and an integer performance level of 32 MIPS. The challenger consists of a standard VME board and one daughter board which together occupy a single VME bus slot. The Challenger includes two processing nodes, each consisting of a TMS320C30 processor, an integrated DMA controller, 256 Kbytes of node-RAM, 32 Kbytes of private RAM, 2 serial links, and a bi-directional parallel port. It also includes 2 to 8 megabytes of parity checked global RAM, a TI to/from IEEE floating point format converter, and a VME master/slave interface. In terms of software, Challenger provides software tools to make full use of the strength and the flexibility of the processors. The tools generate two part programs:

- Host portion: executes on the host CPU, controls and monitors the Challenger's activities, and is the user interface.
- C30 portion: executes the algorithm real-time on the board, making use of the Challenger's parallel processing capabilities.

The Challenger is ideally suited for its role as an intelligent data concentrator. It accepts large quantities of data into its on-board memory from either the host interface or the parallel ports and processes the data in place. The refined results and processed data is then be relayed to the host for post-processing and analysis. In addition to its support of DSP operations, the TMS320C30 can be programmed in a high level language, such as C

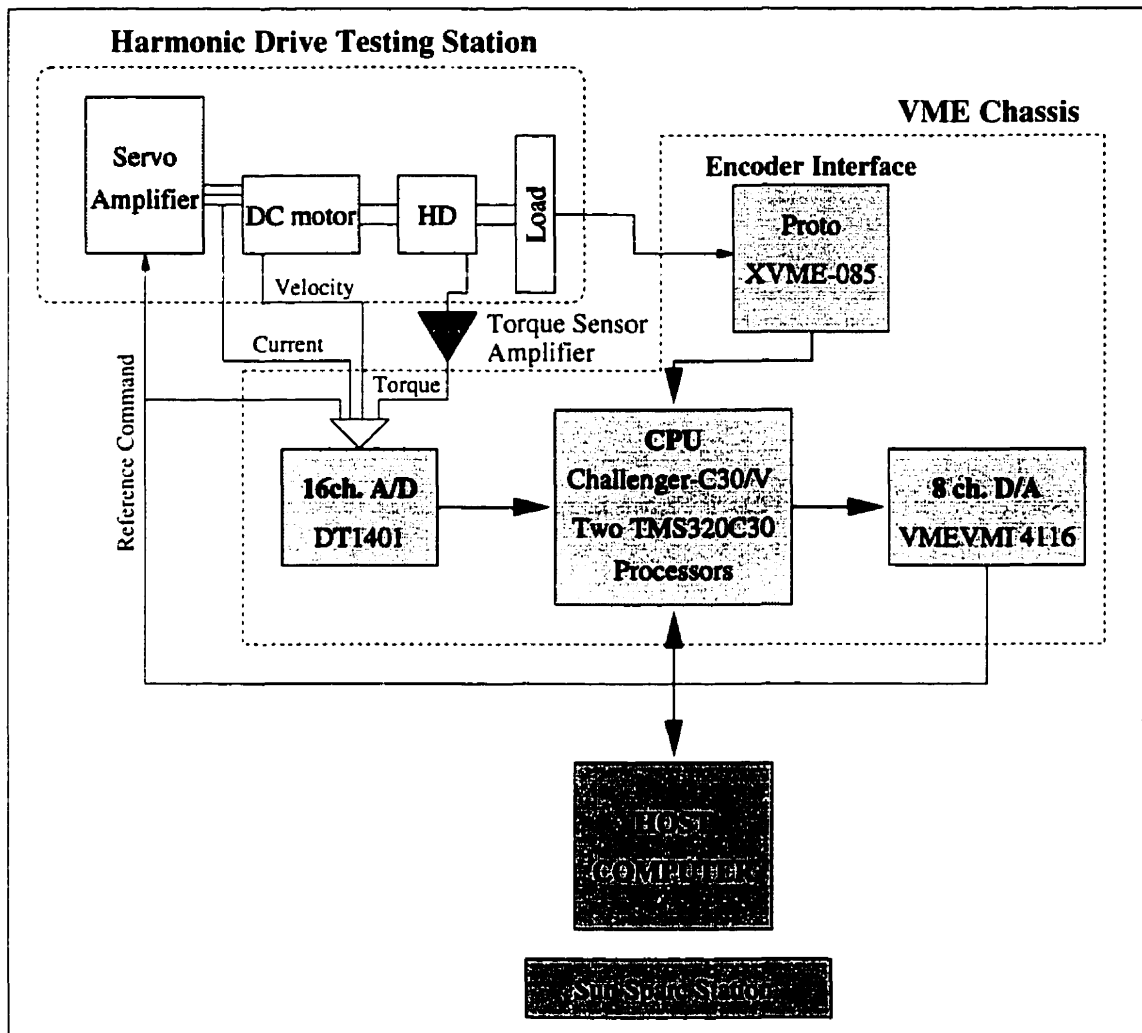


FIGURE 2.12. The schematics of the data acquisition and the processing hardware used for time response experiments

or assembly. The C30 portion of the software, as well as the host portion, were developed on host in C. The compiled version of the code is downloaded into the C30 for on-line implementation of each experiment.

3.2. Other I/O boards' Specifications

The DT1401 is a 16-channels analog-to-digital converter from Data Translation Inc. [59], which receives analog voltages from the sensors and convert them into a 12-bit digital data. The DT1401 is VME-bus compatible I/O board and is mounted on the VME chassis.

It is equipped with a programmable gain amplifier which provides gains of 1, 2, 4, and 8. It accepts both unipolar and bipolar inputs in the range ± 10 V, and it operates at a maximum throughput frequency of 50 kHz. For the dynamic experiments of the harmonic drive system only four channels of the A/D are used.

The VMIVME-4116 is an 8-channel, 16-bit digital-to-analog converter from VME Microsystems International [27]. This board delivers ± 10 V @ 5 mA outputs with positive true offset, binary input coding, and two's complement coding. This board features double buffered data latches, buffered voltage outputs, and selectable external or internal update control strobes. The VMIVME-4116 is also VME-bus compatible and is mounted on the VME chassis. In the harmonic drive experiments only one channel of this board is used, except for the closed-loop frequency response experiments in which two channels are used.

The XVME-085 is the encoder interface housed in the VME Chassis. This interface accepts quadrature and index pulse input signals from the encoder. Four parallel outputs are available for set point triggers. In addition, the starting reference position may be initialized to any given value within the counting range. Home position may also be specified at the user's discretion as the occurrence of the index pulse, a home limit switch input, or as the occurrence of the two in tandem. The interface is completely self-contained and supervises the entire counting and conversion routine after receiving command instructions from the Challenger CPU. Position data is stored on-board until requested by the Challenger. In our setup the encoder reading are logged on board into the Challenger memory.

3.3. Modified Hardware for Frequency Response Experiments

The Siglab hardware is used in frequency response estimates. As explained in § 2 the Siglab is used to generate the excitation signal and to analyse the frequency spectrum of the system. The Siglab could be used stand-alone if no process were required on the output measurements. For harmonic drive experiments however, the flexspline output torque is used for torque feedback application, and as explained in Chapter 3, the harmonic drive torque output displays a high frequency oscillation called torque ripples. Hence, it is necessary for frequency response experiments to filter the measured torque. This process is accomplished by Kalman filter estimation (as elaborated in Chapter 3) of the torque ripples on the Challenger CPU. Figure 2.13 illustrates the modification of the hardware to accommodate the Siglab. The excitation signals are directly applied into the DC motor servo amplifier, while the measurements are processed on-line by the Challenger CPU, and the filtered torque is sent to the Siglab through the D/A converter. The Siglab is linked to the PC host directly, and the frequency response estimate of the system displays on line on the

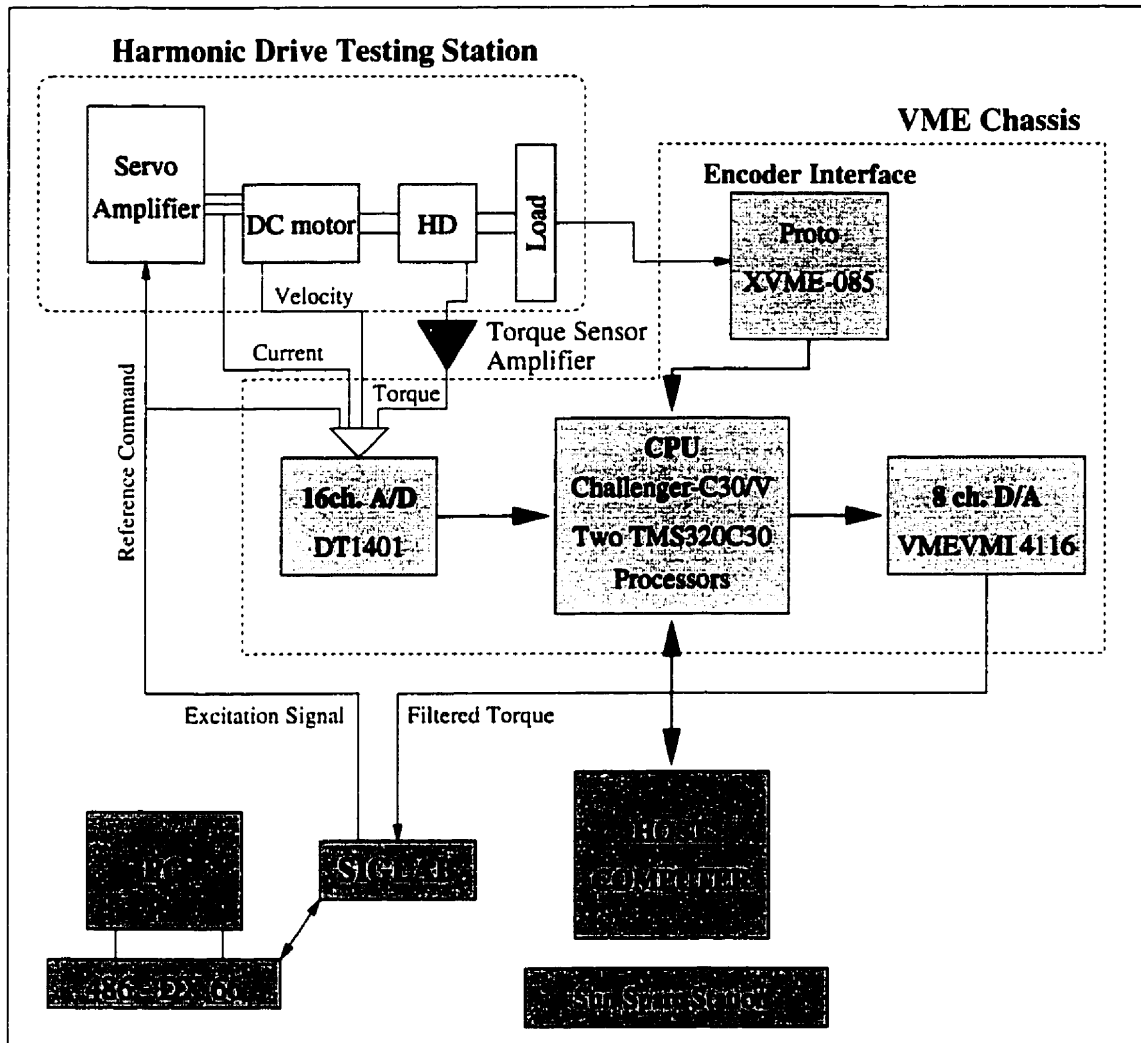


FIGURE 2.13. The schematics of the data acquisition and the processing hardware used for frequency response experiments

PC Monitor while the experiment proceeds. For the experiments where the amplitude of the filtered torques are relatively low, the filtered torque is scaled by a factor of ten (by the Challenger CPU), in order to have a good signal to noise ratio. Moreover, since the record length of the frequency response estimates (especially for sine-sweep experiments) are long, only the experiment data is saved on the PC host for further post-processing and analysis.

4. Software

As explained in § 3.1, the Challenger provides software tools which generates two part programs executed on the host and on the C30 CPU written in C. This section is devoted to introducing the software used for time and frequency response experiments.

4.1. Time Response Software

Figure 2.14 illustrates the flowchart of the software used for open-loop time response experiments, whose program listing is given in § 1 of Appendix B. The software include the programs loaded by the host CPU and by the C30 CPU. In the host portion, the program starts by opening a file to house the experiment result, and by checking if the hardware is ready and free for an experiment. If the hardware is ready the C30 portion of the program is down loaded to the Challenger. For harmonic drive experiments, only one C30 processor of the Challenger is running the main routine of the experiment, and a dummy iteration routine is loaded on the other processor of the Challenger. To generate a reference signal for the experiments, a set of parameters is provided by the operator through keyboard input or a file. These parameters specify the shape function, the DC offset, the amplitude, and the frequency of the reference signal, as well as the sampling period and total experiment time. The shape function available in the software, are sine, square, triangle, inverted sine, half-sine, inverted half-sine, composite-sine (as described in Equation 4.4 of Chapter 4), asymmetric sine, and duty-cycled sine, square and triangle. At this point the loaded programs on the C30 processors will be executed, and the host CPU remains idle, waiting for the real-time C30 process to be terminated. Finally, the program executed on the host CPU will be terminated by fetching the logged Data on the C30 memory and writing it into a file.

The C30 portion of the program executes the main experiment algorithm real-time on board. For each experiment therefore, the details of this part of the program differs to accommodate the experiment details. The structure of the program is however, similar for different experiments. For open-loop time-response experiments as illustrated in Figure 2.14 the code executed by the node 1 of the C30 starts by reading the user-defined parameters from the node RAM, and then initialize the hardware. The main subroutine which synchronizes the fixed-sampling time interrupter executes the given algorithm (procedure c-int10) real-time within the interrupt period. Hence, if the execution time of the algorithm is smaller than the sampling period, the C30 CPU runs idle to synchronize the

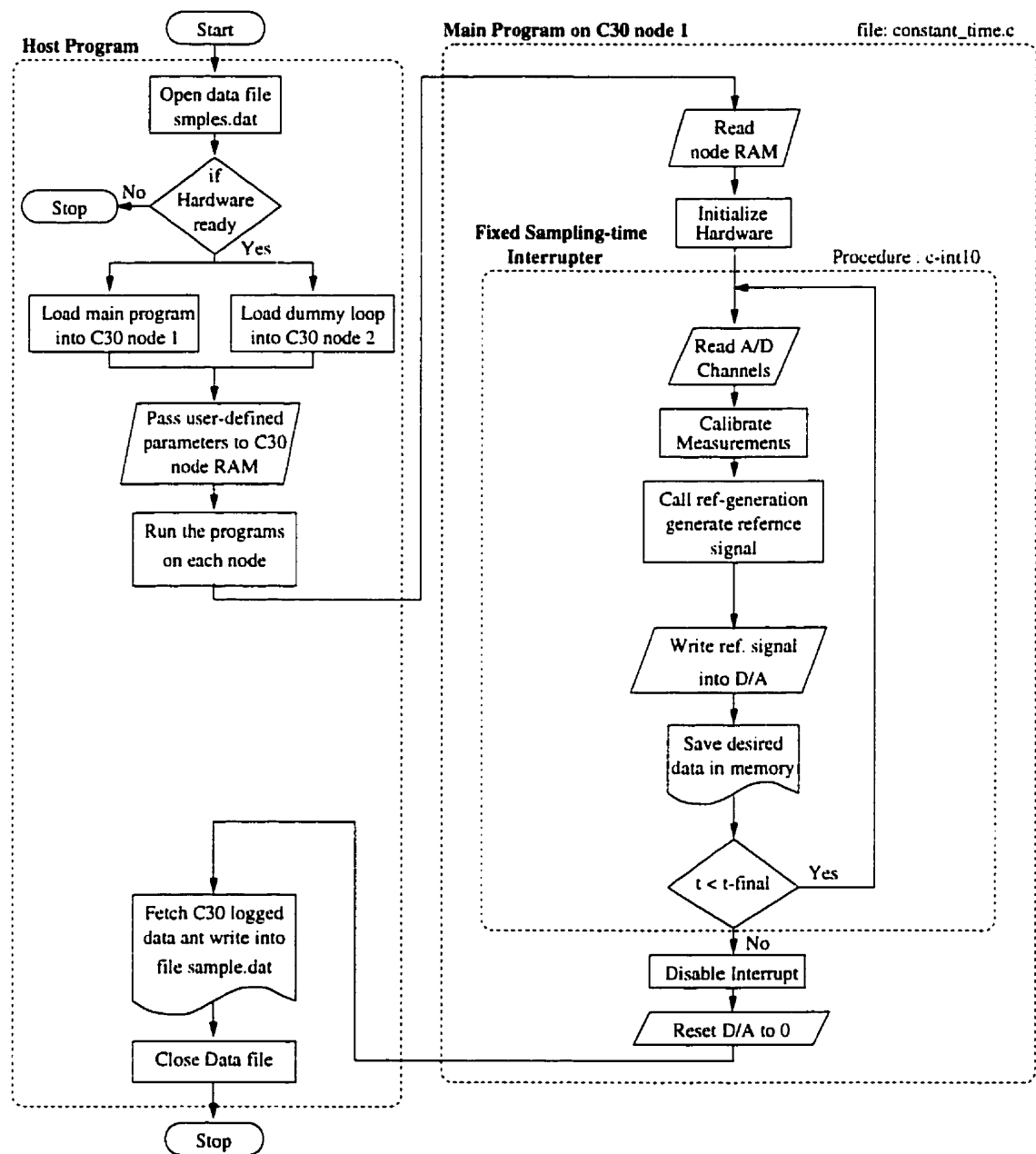


FIGURE 2.14. The flowchart of the typical programs executed on the host and on the C30 CPU for open-loop time response experiments

updating time to the sampling period. Otherwise, it passes a warning to the host, and terminates the execution. For all harmonic drive experiments a sampling period of 1 ms suffices for the execution time of the algorithm in real-time, and avoids experiment interruption.

The algorithm used for open-loop time response experiments has two major components: generating the reference signal and passing it to the hardware, and reading the measurement and saving the calibrated (or uncalibrated) data into memory for off-line post-processing and analysis. The sequence of execution of this algorithm is displayed in the flowchart of Figure 2.14, in which first the measurements are read from the A/D channels and are calibrated, then the reference signal is generated by calling the function *reference-generation* and the value of the reference signal at the current time is written into the D/A. Finally the desired measurements are saved into the C30 memory and the interrupter routine is iterated provided the current time is less than the user-defined final time. After completion of the iteration, the program is terminated by disabling the interrupter, and resetting the D/A to zero.

For closed-loop experiments, the structure of the host and C30 programs are the same as explained before, but the algorithm in the interrupter routine is developed to accommodate the required processes. Figure 2.15 illustrates the flowchart of the host and C30 programs for the most complete case of closed-loop experiments, where a Kalman filter algorithm is implemented to filter the torque ripples, an $\mathcal{H}_\infty$ controller is implemented for the feedback loop and a friction compensation algorithm is implemented in the feedforward loop. The details of torque ripple filter are given in Chapter 3 while the details of $\mathcal{H}_\infty$ controller and friction compensator design and implementation is given in Chapter 6. As illustrated in Figure 2.15 the algorithm executed in the interrupter contains three sub-processes. After reading the data from the A/D channels, and calibrating them, the function *Kalman* is called which estimated the torque ripple at the current time, and the measured torque is filtered using this information. On the other hand, the reference torque signal is generated using *reference-generation* routine, and the torque error is calculated from the subtraction of the filtered torque from the reference signal. By calling the *Control* function the control command with respect to the torque error is calculated by discretized $\mathcal{H}_\infty$ controller. In the meantime the friction torque is estimated by calling the function *Friction* and the corresponding control input is calculated for the friction compensation. Finally, the total control input, which is the addition of the $\mathcal{H}_\infty$ controller input and friction compensation input is written into the D/A to be applied to the DC motor servo amplifier. The rest of

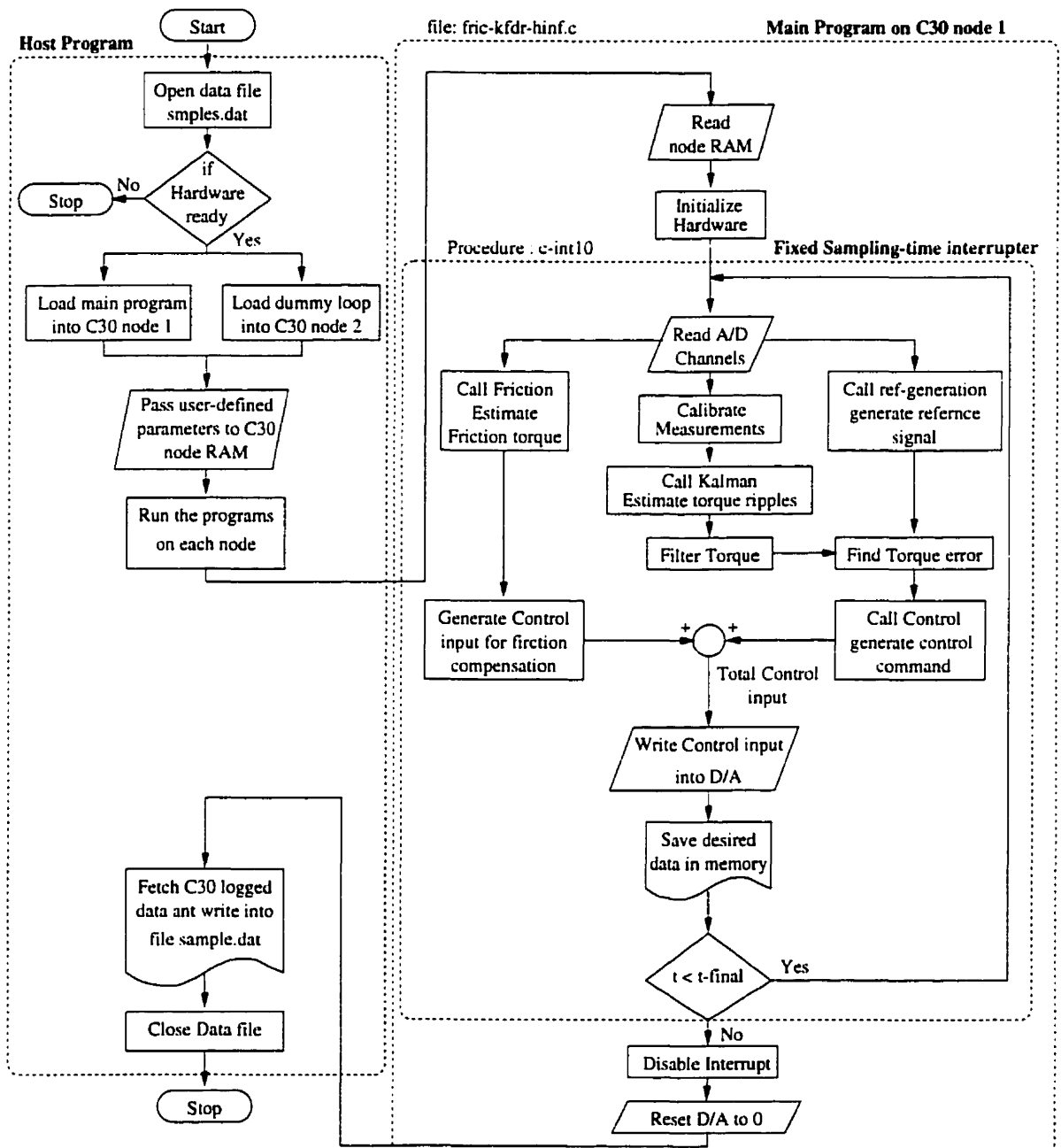


FIGURE 2.15. The flowchart of the typical programs executed on the host and on the C30 CPU for closed-loop time response experiments

the program is similar to that in open-loop experiments. A listing of the C30 portion of the closed-loop time response experiment is given in § 4 of Appendix B.

4.2. Frequency Response Software

In frequency response experiments Siglab hardware is used to generate the excitation signal and to analyse the frequency spectrum of the system. The software is slightly modified to accommodate this changes in the hardware. Figure 2.16 illustrates the flowchart of the software used for closed-loop frequency response experiments. In the host portion of the software the parameters defined by the user are reduced to the sampling frequency, and the total time of the experiments. The parameters related to the reference signal are not passed to C30 node RAM, since the software is not generating the reference signal. In the C30 portion of the software, the reference signal generated by the Siglab hardware is read through A/D channel 1, and not only the control input is fed to the amplifier through D/A channel 1, but also the filtered torque is written in D/A channel 2 to be used by Siglab for on-line frequency response estimation of the system. The other difference in the software structure is due to the fact that the measurements are not saved in the Challenger memory for further use. Therefore, in the host portion of the program no data is fetched from the Challenger memory.

The software used for open-loop frequency response experiments is similar in structure to that given in Figure 2.16, with the only difference that neither friction compensation, nor control routine are available, and the reference signal generated by the Siglab is passed directly to DC motor servo amplifier. A listing of the code used for open-loop and closed-loop frequency response experiments are given in § 5, and § 6 of Appendix B. respectively.

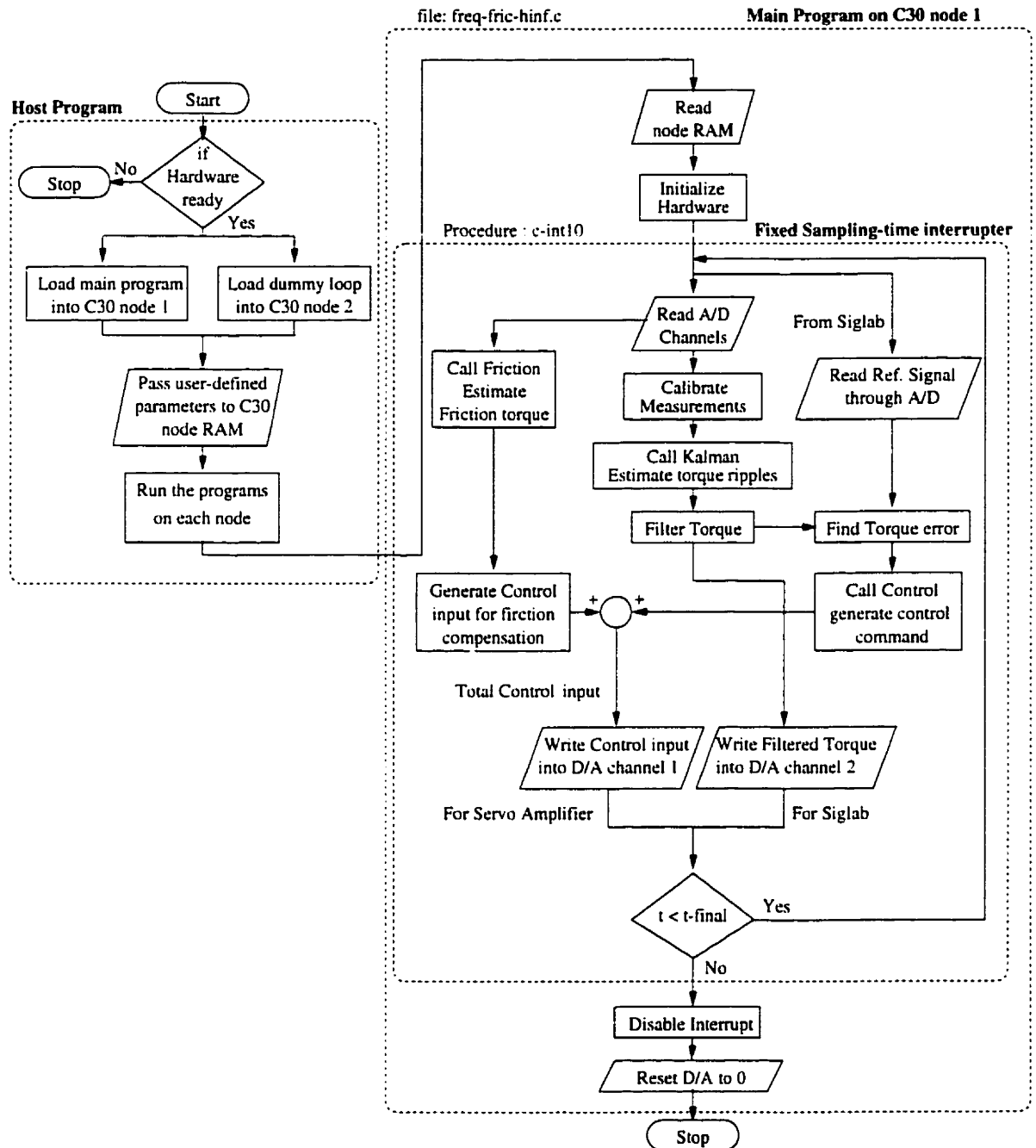


FIGURE 2.16. The flowchart of the typical programs executed on the host and on the C30 CPU for closed-loop frequency response experiments

CHAPTER 3

Intelligent Built-in Torque Sensor

In order to apply torque feedback on the robot joint, it is necessary to measure the transmitted torque through the actuator transmission mechanism. Conventionally, torque sensor are placed in the output transmission line of the robot [58, 101]. However, for a harmonic drive transmission, which has an elastic element named flexspline, there is no advantage to add an additional compliant element and thereby reduce the joint stiffness. This chapter is devoted to the design and implementation of a built-in torque sensor for harmonic drives as first proposed by Hashimoto in 1989 [49]. This method proved to be an economical and effective way of torque sensing for harmonic drives in our setup, as claimed by Hashimoto et al. [51]. In our testing station a Wheatstone bridge of four Rosette strain-gauges is utilized to sense the torsional torque transmitted through the flexspline. A practical and accurate method is proposed to mount the strain-gauges on the flexspline in order to minimize any radial or circumferential misplacement of each strain-gauge.

One important characteristic of harmonic drive torque transmission, as observed in free motion experiments, is a high frequency oscillation in the output torque signal. These oscillations named torque ripples whose principal frequency of oscillation (in rad/sec) is twice the motor velocity (in rad/sec), are mainly caused by harmonic drive gear meshing vibration. A small fraction of the torque ripples are caused by the non-ideal torque measurement, because of the direct attachment of strain gauges on the flexspline. Since the flexspline has an elliptical shape, strain gauges mounted on the flexspline are subjected to unwanted strain caused by the elliptical shape. We show in this chapter, however, that using four Rosette strain-gauges, and using an accurate method to mount the strain-gauges, will reduce the amplitude of the torque ripple to a minimum. Moreover, the dependence of the frequency content of the torque ripples on the velocity makes it possible to model them as a simple harmonic oscillator, and to employ a Kalman filter to predict and filter

them from the torque measurement. If only low-frequency torque control is desired [105], the high-frequency ripples may be removed by estimating them via Kalman filtering. This is more efficient than simple low-pass filtering, because it uses the known structure of the torque signal.

We proposed a fourth-order harmonic oscillator error model is proposed in this chapter to characterize both the fundamental and first-harmonic frequency content of the torque ripple. Using this model, a prediction-type Kalman filter algorithm is applied to estimate the torque ripples. The performance of the on-line implemented Kalman filter for torque ripple cancellation is shown to be quite fast and accurate.

The Kalman filter is used not only to estimate the torque ripples, but also to cancel any mechanical misalignment torque signature on the measured torque. Many torque sensors exhibit the limitation of being sensitive to the torques applied on the direction perpendicular to their axis of measurement. In our setup after repeated use of the harmonic drive system for different experiments, a similar torque signature was observed on the measured torque. After examining the system accurately, the source of this torque signature is found to be the misalignment of the harmonic drive shaft and the load. The frequency of misalignment torque (in rad/sec), as it can be intuitively identified from its source of generation, is found to be exactly the same as the output shaft velocity (in rad/sec). Therefore, by adding another block to the harmonic oscillator model we are able to estimate the misalignment component of the measured torque. By using the extended model for the torque ripple and misalignment torque together, it is shown that the filtered torque cancels the torque ripples and misalignment torque quite accurately. The experimental results obtained from combining the original ideas introduced in this chapter with robust torque controller design given in Chapter 6 and in [104], proves that built-in torque sensors are viable and economical means to measure harmonic drive transmitted torque and to employ them for torque feedback strategies.

1. Built-In Torque Sensor

As illustrated in Figure 3.1 four Rosette strain-gauges are mounted on the diaphragm part of the flexspline. A Rosette strain-gauge consist of two separate strain-gauges perpendicularly mounted on one pad. For a clockwise torque exerted on the flexspline illustrated in Figure 3.1 strain-gauge R_1 is under compression while strain-gauge R_2 is under tension. Similarly all odd-indexed strain-gauges are under compression, while the others are

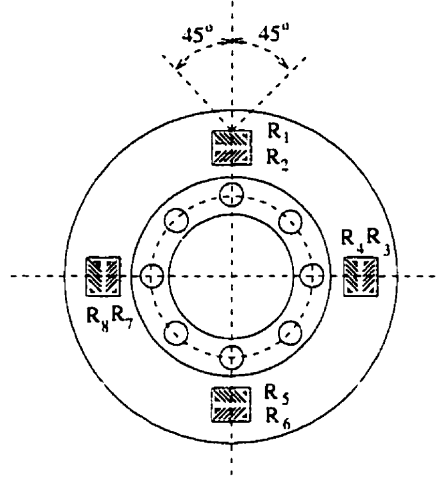


FIGURE 3.1. Position of four Rosette strain-gauges on the flexspline diaphragm

under tension. Thus, a Wheatstone bridge of strain-gauges as illustrated in Figure 3.2 can transduce the torsion into a difference voltage. The reason why Rosette strain-gauges are necessary for harmonic drive built in torque sensor, is to compensate for the elliptical shape of the flexspline. If the strain caused by applied torque is named ϵ_t , while the strain caused by the elliptical shape of flexspline is called ϵ_w , Hashimoto et al. [52] illustrated that the strain applied to strain-gauge R_1 , and R_2 are

$$\begin{cases} \epsilon_1 = \epsilon_t + \epsilon_w \\ \epsilon_2 = -\epsilon_t + \epsilon_w' \end{cases} \quad (3.1)$$

where ϵ_w' is assumed in [51] to be a sinusoidal modulation of ϵ_w . Thus,

$$\epsilon_1 - \epsilon_2 = 2\epsilon_t + \Psi_0 \sin(2\beta) \quad (3.2)$$

In order to cancel the modulation $\Psi_0 \sin(2\beta)$ and to detect the actual torsional strain ϵ_t the information from strain-gauges R_3 , and R_4 is necessary. These strain-gauges are located at an angle of 90° from Strain-gauges R_1 and R_2 , and therefore:

$$\epsilon_3 - \epsilon_4 = 2\epsilon_t + \Psi_0 \sin(2\beta - \pi) \quad (3.3)$$

Therefore, a Wheatstone bridge, constructed from strain-gauges R_1 to R_4 is sufficient to produce a difference voltage proportional to the torsional strain ϵ_t as following

$$\begin{aligned} E_{out} &= \frac{K}{4}(\epsilon_1 + \epsilon_3 - \epsilon_2 - \epsilon_4)E_{sup} \\ &= K\epsilon_t E_{sup} \end{aligned} \quad (3.4)$$

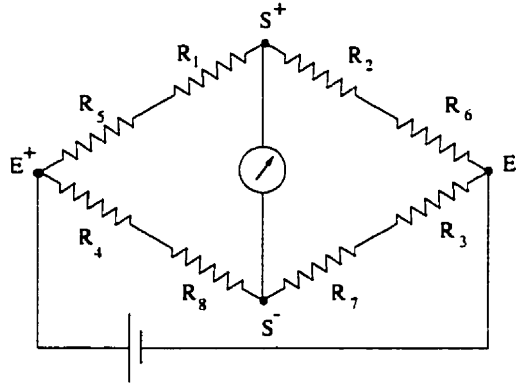


FIGURE 3.2. Strain-gauges Wheatstone bridge network

in which K is the gauge factor, and E_{out} and E_{sup} denote output and supply voltages, respectively.

1.1. Positioning Error of Strain-gauges

Although, only two Rosette strain-gauges are sufficient to extract the torsional strain, two other Rosette strain-gauges are introduced to maintain symmetry, and to minimize the effect of positioning error. Sensing inaccuracies are caused by radial, circumferential, and angular positioning error. Radial positioning error occurs when the gauges are placed on different radii from the center of the flexspline. As shown in Figure 3.1 even without any radial misplacement of the strain-gauges, radial error exists, since strain-gauges R_1 and R_2 are placed at different radii. However, by using four Rosette strain-gauge, as illustrated in Figure 3.1, this error will be compensated by strain-gauges R_5 , and R_6 which are located in the reverse position.

Circumferential positioning error occurs when two Rosette strain-gauges are mounted in an angle different than 90° from each other. This positioning error introduces more sensing inaccuracies than radial positioning error [52, 108]. Angular positioning error occurs when a Rosette strain-gauge is not mounted perpendicular to the flexspline's axis of rotation. This positioning error also introduces sensing inaccuracy similar to the circumferential positioning error. To minimize the positioning error, we propose a method using a specially-designed transparent film for the strain-gauges placement. As illustrated in Figure 3.3, an accurate drawing of the strain-gauge placement positions is printed on a transparent film using a laser printer with the finest possible lines. Then the strain-gauges are placed on the transparent film using a microscope. By this means all positioning errors are reduced to a minimum, and as examined in our testing station to other placement

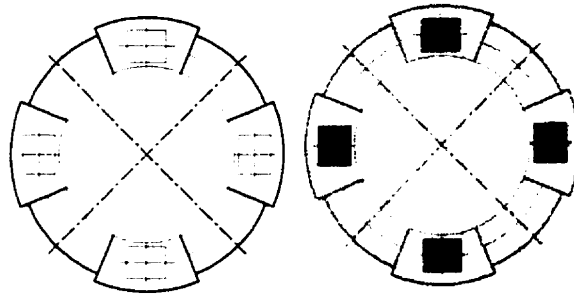


FIGURE 3.3. Proposed transparent film for accurate strain-gauge placement

methods, the sensing accuracy is maximized. When the strain-gauges are mounted on the transparent film as illustrated in Figure 3.3, the transparent film is placed on the flexspline and the strain-gauges are cemented on the surface. The final configuration of the wired strain-gauges are illustrated in Figure 3.4. The harmonic drive used in the testing station and illustrated in Figure 3.4 is from RHS series of HD systems, with gear ratio of 100:1, and rated torque of 40 Nm . The flexspline diameter is 46 mm, and its length is 49 mm. To amplify the output signal, a variable range amplifier with gain 1000 is used, while it provides 10 volts DC voltage as input voltage to the Wheatstone bridge.

2. Torque Sensor Calibration

To examine the dynamics of the torque sensor and to calibrate it, we locked the harmonic drive wave generator, and applied a known torque on the flexspline. The locking device is a simple shaft resembling the motor shaft, which can be fixed to the ground. The

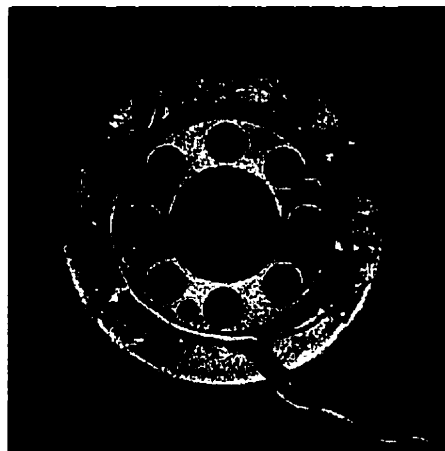


FIGURE 3.4. Harmonic drive built-in torque sensor

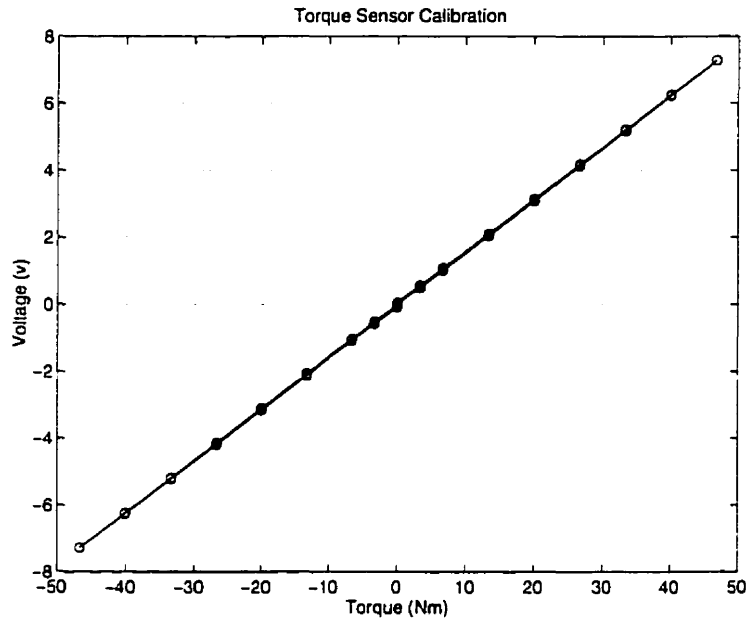


FIGURE 3.5. Torque sensor calibration results for full-range loading

output torque is applied using arm and weight method. The torque calibration consists of loading and unloading some weight on the arm, in both direction. This experiment is repeated at different flexspline positions, to check the position dependence of the torque sensor. Also low-range and full-range loading experiments are tested at each position to evaluate the nonlinearity of the torque sensor. A typical result for full-range loading experiment is shown in Figure 3.5. These data are best fitted by a straight line using least-squares approximation. The estimated torque sensor gain for each experiment is calculated from the slope of this line. However, this gain is deemed acceptable, only if it is consistent for other experiments. By consistency we mean a statistical measure, namely *the ratio of the standard deviation to the average value of estimated parameter for different experiments* [103]. If this measure is small, we have a good consistency for different experiments, and the final gain is obtained from the average value of the estimated gain for different experiments. The final gain is obtained by this method for eight experiments, and the results are given in Table 3.1, where the gain is 6.6 Nm/volts, with a low consistency measure of 2%. This

TABLE 3.1. Torque sensor calibration results

Torque Sensor Gain	Consistency Measure
6.60 Nm/Volts	2%

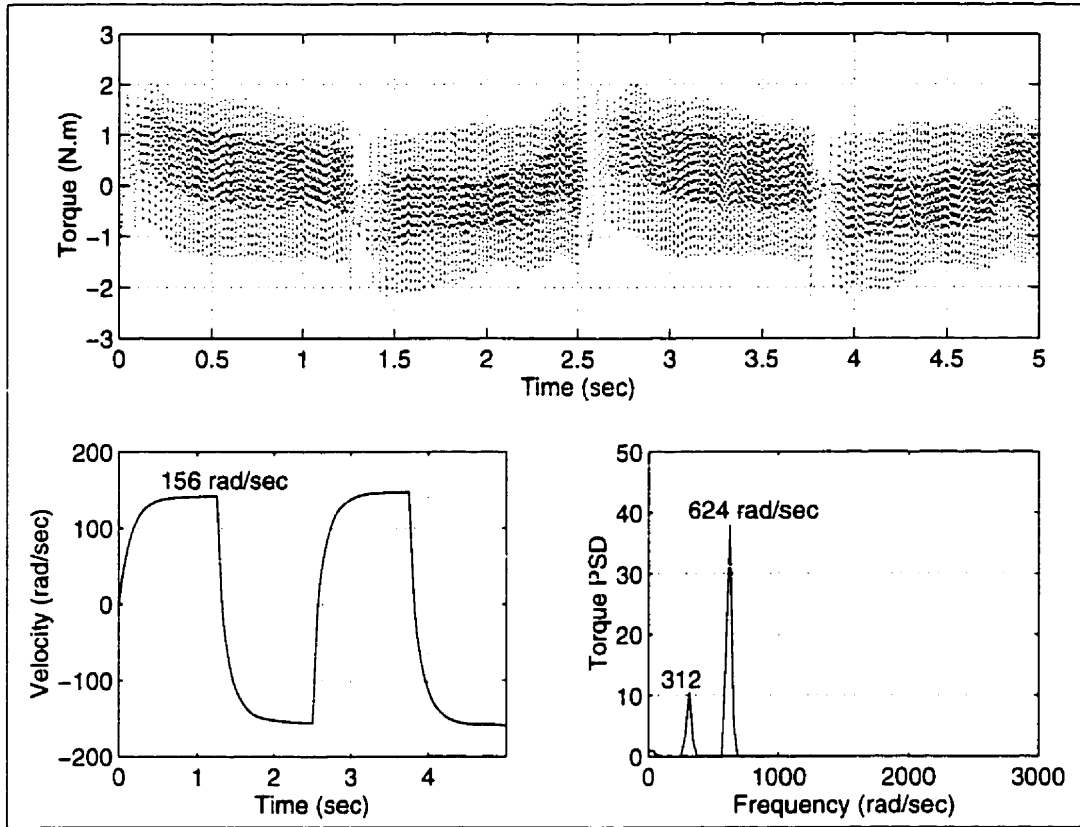


FIGURE 3.6. Measured Torque and velocity and the power spectrum of the measured torque with peaks at multiples of the velocity

illustrates that the torque sensor is quite linear, and it is not position dependent. Moreover, the torque sensor gain is consistent for different operating conditions.

3. Torque Ripple Compensation

One important characteristic of harmonic drive torque transmission as observed in free motion experiments, is a high frequency oscillation in output torque signal (See torque curve in Figure 3.6). These oscillation, named torque ripples, were also observed by other researchers [42]. Torque ripples are caused mainly by harmonic drive gear meshing vibration. Harmonic drive gear meshing vibration introduces a real torque oscillation which can be observed in the end effector motions of robots using harmonic drives and even sensed by hand when back-driving the harmonic drive. Its principal frequency of oscillation (in rad/sec) is twice the motor velocity (in rad/sec), for the gear teeth in harmonic drives are meshing in two zones. A small fraction of the torque ripples are caused by the non-ideal torque

measurement, because of the direct attachment of strain gauges on the flexspline. Since the flexspline has an elliptical shape, strain gauges mounted on the flexspline are subjected to unwanted strain caused by the elliptical shape. Hashimoto [51], proposed using at least two pairs of Rosette strain gauges to compensate for this unwanted strain. However, ideal compensation is possible only if there is no positioning error of strain-gauges. As explained in § 1 it has been shown however, that using four Rosette strain-gauges, and using an accurate method to mount the strain-gauges will reduce the amplitude of the torque ripple to a minimum. Unfortunately the frequency of torque ripples (in rad/sec) introduced by the non-ideal behavior of the sensor is also twice the motor speed (in rad/sec), since the major axis of the ellipse is travelling twice as fast as the wave generator. This make it impossible to discern the true ripples caused by the gear meshing vibration from that caused by non-ideal measurement. As illustrated in Figure 3.6, the power spectrum of the measured torque plotted for the time interval 0.8 to 1.1 sec when the velocity is almost flat and about 156 rad/sec shows two peaks at 312, and 624 rad/sec. This confirms the existence of the fundamental frequency of the oscillation as twice the velocity and shows the significance of the next important first-harmonic frequency of four times the velocity. The dependence of the frequency content of the torque ripples on the velocity makes it possible to model them as a simple harmonic oscillator, and to employ a Kalman filter to predict and filter them from the torque measurement. If only low-frequency torque control is desired [105], the high-frequency ripples may be removed by estimating them via Kalman filtering. This is more efficient than simple low-pass filtering, because it uses the known structure of the torque signal.

3.1. Torque Ripple Model

A fourth-order harmonic oscillator error model can characterize both the fundamental and first-harmonic frequency content of the torque ripple. This can be represented by the following discrete state space form:

$$\begin{cases} \mathbf{x}(k+1) = \left[\begin{array}{c|c} \Phi_1(k) & \mathbf{0} \\ \hline \mathbf{0} & \Phi_2(k) \end{array} \right] \mathbf{x}(k) + \mathbf{w}(k) \\ y(k) = [1 \ 0 \ 1 \ 0] \mathbf{x}(k) + v(k) \end{cases} \quad (3.5)$$

where

$$\Phi_i(k) = \begin{bmatrix} \cos(\omega_i(k)T_s) & \sin(\omega_i(k)T_s) \\ -\sin(\omega_i(k)T_s) & \cos(\omega_i(k)T_s) \end{bmatrix} \quad (3.6)$$

in which T_s is the sampling period, and $\omega_1(k) = 2\dot{\theta}(k)$, $\omega_2(k) = 4\dot{\theta}(k)$, and $\dot{\theta}(k)$ is the motor shaft velocity in rad/sec at time step k . Moreover, y is the torque ripple, which needs to be observed from the torque measurement by having a crude model for the expected torque. If measured torque is indicated by T_{meas} , and the expected torque is indicated by T_{exp} then torque ripple $y(k) = T_{meas}(k) - T_{exp}(k)$, while the difference between true torque and the crude estimate of the expected torque is encapsulated by the measurement errors in $v(k)$ in Equation 3.5. Therefore, no accurate model for the expected torque T_{exp} is necessary and hence, for free motion experiments the expected torque can be estimated simply by the inertial part of the output torque which is the load inertia multiplied to the load acceleration. The first two elements of the state $\mathbf{x}$ consists of the component of the torque ripple due to the fundamental frequency ω_1 , and its derivative, while the next two elements are those components of torque ripple due to the first-harmonic frequency ω_2 , and its derivative. Therefore, the total torque ripple is calculated by adding the first and third element of the state. $\mathbf{w}(k)$ characterizes the other frequency components of the torque ripples which will not be estimated in this model.

3.2. Torque Ripple Estimation

Using the fourth-order harmonic oscillator model for the torque ripples, a prediction-type Kalman filter algorithm is applied to estimate the torque ripples [90]. A Prediction-type estimate is computationally faster than a current-estimate type of Kalman filter, and therefore preferable for online implementation. Assuming that the process noise $\mathbf{w}(k)$ and measurement noise $v(k)$ are zero-mean Gaussian white and have covariances defined by $\mathbf{Q}$ and R as:

$$E\{\mathbf{w}(k)\mathbf{w}^T(k)\} = \mathbf{Q} \quad , \quad E\{v(k)v^T(k)\} = R \quad (3.7)$$

the states estimates is calculated using kalman filter formulation [90] as following:

$$\hat{\mathbf{x}}(k+1) = \Phi(k)\hat{\mathbf{x}}(k) + \mathbf{K}_e(k) [\mathbf{y}(k) - \mathbf{C}(k)\hat{\mathbf{x}}(k)] \quad (3.8)$$

in which the kalman filter gain $\mathbf{K}_e(k)$ will be updated from,

$$\mathbf{K}_e(k) = \Phi(k)\mathbf{P}(k)\mathbf{C}^T(k) [R(k) + \mathbf{C}(k)\mathbf{P}(k)\mathbf{C}^T(k)]^{-1} \quad (3.9)$$

$$\mathbf{P}(k+1) = \mathbf{Q}(k) + [\Phi(k) - \mathbf{K}_e(k)\mathbf{C}(k)] \mathbf{P}(k)\Phi^T(k) \quad (3.10)$$

Since the measurement signal $y(k)$ is a scalar, its covariance matrix $R(k)$ is also a scalar, and no matrix inversion is required for online implementation of the kalman filter gain given in Equation 3.9. Figure 3.7 illustrates the performance of the Kalman filter implemented

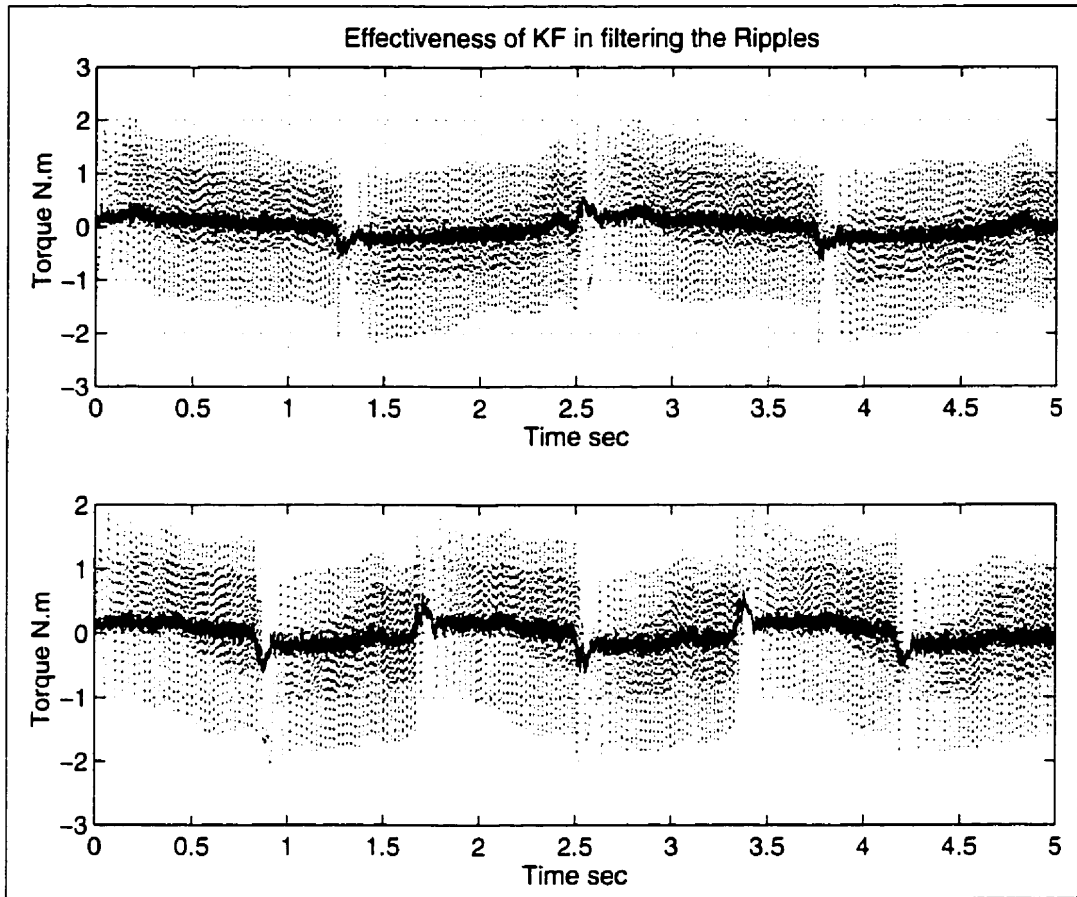


FIGURE 3.7. Kalman filter performance to cancel torque ripples. Dotted: Measured torque, Solid: Kalman filtered torque

on line with a sampling frequency of 1 kHz to estimate and filter the torque ripples for two typical experiments, in which $Q = 10^3 I_{4 \times 4}$ and $R = 1$. The performance of the Kalman filter for torque ripple cancellation is shown to be quite fast and accurate. The Software code which implements the Kalman filter on-line is listed in § 2 of Appendix B.

4. Misalignment Torque Compensation

The Kalman filter can be used not only to estimate the torque ripples, but also to cancel any mechanical misalignment torque signature on the measured torque. Many torque sensors exhibit the limitation of being sensitive to the torques applied on the direction perpendicular to their axis of measurement. In our setup after repeated use of the harmonic drive system for different experiments, a similar torque signature was observed on the measured torque. Figure 3.8 illustrates a simple experiment in which the harmonic drive

is driven by a constant velocity but the measured torque exhibit a sinusoidal trend. The expected torque, the solid line in Figure 3.8, is constant after a short time of acceleration, but the measured torque, dotted lines, displays a sinusoidal behavior. After examining the system accurately, the source of this torque signature is found to be the misalignment of the harmonic drive shaft and the load. By disassembling the system and carefully reassembling it, the peak to peak amplitude of this misalignment signature was reduced from 10 N.m to less than 2 N.m. However, in practice it is quite expensive, and probably infeasible to perfectly align all the moving components. Fortunately, the sinusoidal feature of this misalignment torque makes it possible to accurately estimate them with an error model. The frequency of misalignment torque (in rad/sec), as it can be intuitively identified from its source of generation, is exactly the same as the output shaft velocity (in rad/sec). Therefore, adding another block to the harmonic oscillator model (given in Equation 3.5) of the system with frequency $\omega_3(k) = vel(k)/(Gear\ Ratio)$, will estimate the misalignment component of the measured torque. Using this sixth order model for the torque ripple and misalignment torque together, Figure 3.9 illustrates the Kalman filter performance in cancelling those elements for two typical experiments. The Kalman filtered torque is shown to cancel the torque ripples and misalignment torque quite accurately. These results are obtained using an online Kalman filter implementation on the system with sampling frequency of 1 kHz. § 3 of Appendix B gives the program listing of this algorithm. In contrary to low-pass filtering which is incapable of extracting the misalignment torque from the measurements, Kalman filter estimation proved to be fast and accurate to filter both torque ripples and

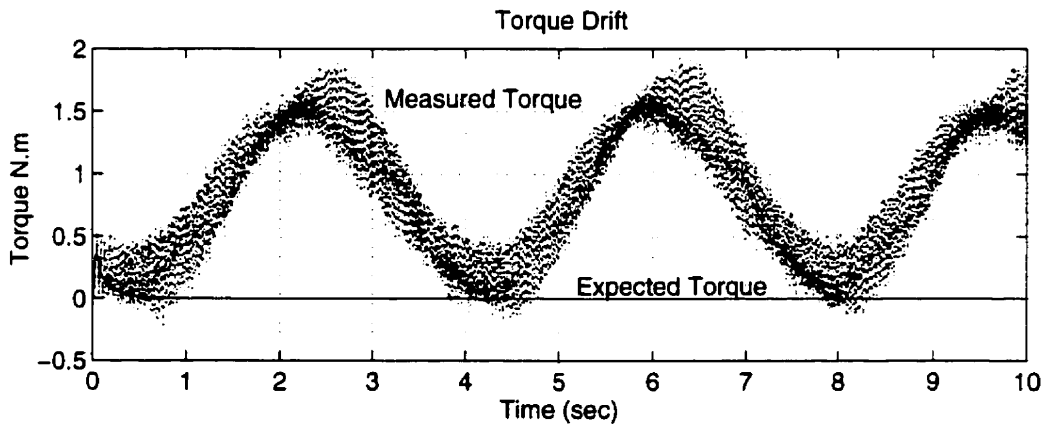


FIGURE 3.8. Misalignment torque signature on the measured torque

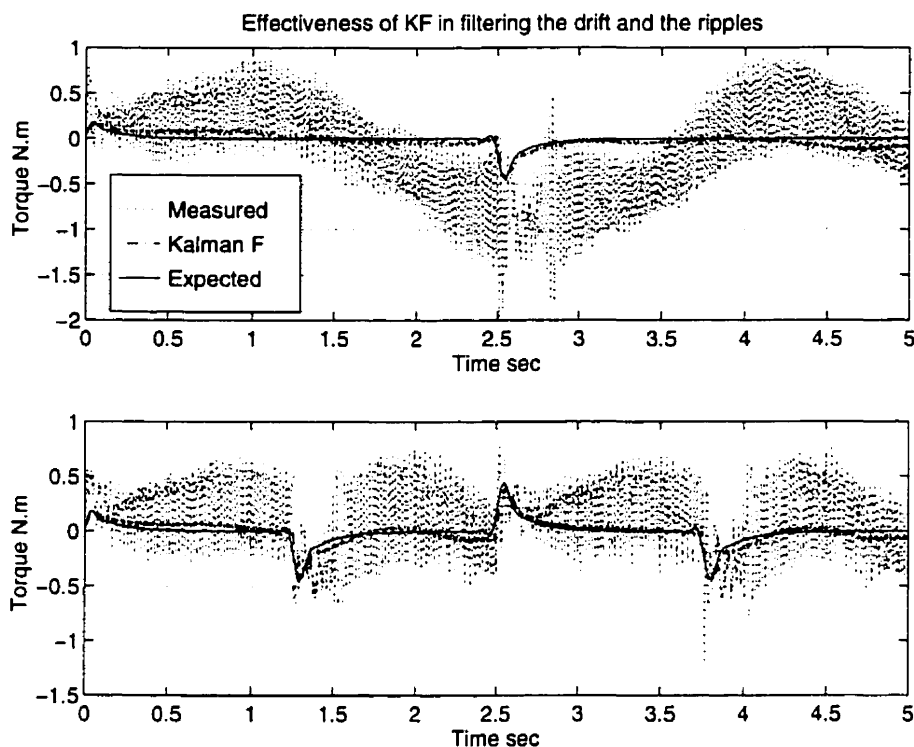


FIGURE 3.9. Kalman filter performance to cancel torque ripples, and misalignment torque for two typical experiments

misalignment torque. Moreover, it is a reliable method for different operating ranges, and therefore, preferable for torque feedback.

5. Summary

In this chapter the built-in torque sensor for harmonic drive systems as first proposed by Hashimoto is examined in detail. By this method strain-gauges are directly mounted on the flexspline, and therefore, no extra flexible element is introduced into the system. To have minimum sensing inaccuracy, four Rosette strain gauges are employed using an accurate method of positioning. An accurate drawing of the strain-gauge placement positions is printed on a transparent film, and the strain-gauges are placed on the transparent film using a microscope. Then the transparent film is accurately placed on the flexspline, and the strain-gauges are cemented on the surface. It is shown that employing this method reduces the positioning error to a minimum. Calibration of the torque sensor shows that the sensor is performing linearly and the torque readings are not dependent to the position of the flexspline.

One important characteristic of harmonic drive torque transmission, as observed in free motion experiments, is a high frequency oscillation in the output torque signal. To cancel these oscillations named torque ripples from the measured torque, Kalman filter estimation is employed. Due to the dependence of the frequency content of the torque ripples on the wave generator velocity, a simple fourth order harmonic oscillator proved to accurately model the torque ripples. The performance of Kalman filter to cancel the torque ripples from torque measurements is shown to be very fast and accurate. Moreover, the error model is extended to incorporate any misalignment torque signature. By on line implementation of the Kalman filter incorporating a sixth order model, it is shown that this method is a fast and accurate way to filter torque ripples and misalignment torque, and hence, this intelligent built-in torque sensor is preferable for torque feedback. This method is implemented as an integral part of the torque feedback algorithms proposed in Chapter 6.

CHAPTER 4

Modelling and Identification

The goal of modelling the harmonic drive system is to discover the simplest representation which can replicate system behaviour to a desired level of accuracy. Our particular goal in modelling is to ultimately implement a model-based torque control algorithm on the system. Moreover, we used the computer model for examining and improving control laws, prior to implementing them. As recommended by other researchers [76, 121], in order to achieve this objective, it is necessary to have at least a simple and accurate model for friction and compliance of harmonic drive systems. In practice it has been proven that the knowledge obtained through the process of modelling and identification a powerful medium for understanding and improving the design, as well as for providing new horizons for controller design.

In this chapter a moderately complete model for harmonic drive systems is developed. A nonlinear model for the DC motor is proposed which includes Coulomb and viscous friction models. An empirical linear model for the servo amplifier is developed next, using experimental frequency response estimates. The harmonic drive is represented as a two-input two-output system which suffers from nonlinear compliance and friction. It is shown that a linear model for stiffness combined with a velocity-dependent structural damping model can replicate the hysteresis torsion curve of the system compliance. The frictional losses of the transmission are modelled using Coulomb friction, viscous damping and Stribeck friction. Both high speed and low speed friction terms have been identified using constrained and free motion experiments respectively.

1. DC Motor

A DC motor can be viewed as a two-input, one-output black-box, where the servo current and external torque are the inputs and the angular displacement (or velocity) is the

output. The torque balance for the DC motor can be written in the form:

$$K_m i = J_m \ddot{\theta} + T_{f_m} + T_{out} \quad (4.1)$$

where K_m is the motor torque constant, i is the input current to the motor, J_m is the motor inertia, and T_{out} is the external torque. T_{f_m} is the friction torque, which can be modelled in the form of velocity direction dependent viscous and Coulomb friction via:

$$T_{f_m}(\dot{\theta}) = T_{v_n} u_{-1}(-\dot{\theta})\dot{\theta} + T_{v_p} u_{-1}(\dot{\theta})\dot{\theta} + T_{s_n} u_{-1}(-\dot{\theta})\text{sign}(\dot{\theta}) + T_{s_p} u_{-1}(\dot{\theta})\text{sign}(\dot{\theta}) \quad (4.2)$$

where

$$u_{-1}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \quad (4.3)$$

Note that the indices p and n represents the dependence of the friction coefficients on the velocity direction.

The model parameters are estimated by least-squares approximation as explained in § 1.1 of Appendix A. Using a linear regression model, and measuring the motor velocity and current for two sets of high- and low-velocity experiments, the model parameters can be estimated using the *Moore-Penrose generalized inverse* [13, 34]. Householder reflection is utilized in numerical calculations to avoid ill-conditioning [44].

Four types of inputs are applied to the servo amplifier, and motor velocity and current are measured and logged in each experiment. The input shape functions are sinusoidal, square, triangular, and composite sine waves, where the composite sine waves are the superposition of three sinusoids as,

$$\cos(\omega t) + 0.5\sin(2\omega t) + \sin(3\omega t) \quad (4.4)$$

For each signal type, frequencies from 0 to 10 Hz and amplitude from 10 % to 100 % of maximum allowable amplitude are spanned, to experiment both low and high velocities with wide frequency range. To generate the motor acceleration from the velocity signal, first the velocity signal is filtered using a fifth-order Butterworth filter, by zero-phase distortion routine, and then numerically differentiated. The details and advantages of this method are analyzed in [102].

By means of least-squares estimation, for each experiment we obtain a set of parameters for the linear regression model of Equation 4.1. However, these parameters are deemed acceptable only if they are consistent between experiments. This can be quantified by a statistical measure, namely *the ratio of the standard deviation to the average value of each parameter estimated for different experiments* which is called here the *consistency measure*.

TABLE 4.1. DC motors estimated parameters

	McGill DC Motor		ISE DC Motor	
	Estimated Parameter	Consistency Measure	Estimated Parameter	Consistency Measure
K_m	0.0542	0%	0.1815	0%
J_m	5.5×10^{-5}	0.82%	5.8×10^{-5}	5.75%
T_{vp}	5.3×10^{-5}	23.3%	8.6×10^{-4}	7.75%
T_{vn}	5.3×10^{-5}	11.5%	5.8×10^{-4}	14.4%
T_{sp}	1.4×10^{-2}	9.01%	1.5×10^{-2}	25.9%
T_{sn}	1.3×10^{-2}	5.72%	2.7×10^{-2}	25.8%

If the consistency measure is small, we have consistency for different experiments, and the model is capable of capturing the dynamics of the system.

Figure 4.1 illustrates the velocity fit obtained by the model for two typical experiments. The model is able to capture the dynamics of the system for both low and high velocity experiments. Table 4.1 summarizes the estimation results for two setup DC motors, where for each setup about 15 experiments are considered (All parameters expressed in SI units). The motor torque constant K_m is obtained from the motor specs given in Table 2.1, and is assumed to be known in the least-squares estimation. The consistency measure of the results for different experiments shows to be less than 10% for some parameters, and less than 30% in others. The reason for variation in consistency measure among parameters is that the responses are relatively insensitive to variations in some parameters. However, as illustrated in Figure 4.1 and in [102], it has been verified by simulations that having consistency measure less than 30% gives a relatively good match to the experiments.

2. Servo-Amplifier

The structure of the servo amplifier and its operating modes are explained in § 1.1 of Chapter 2. Ideally, the servo amplifier acts as a current source for the DC motor, when operated in “torque mode”. Hence, the output current of the servo amplifier is expected to be proportional to its commanded voltage. Servo amplifiers suffer in practice, however, from limited bandwidth, and output current saturation. To capture the dynamics of the servo amplifier, the frequency response estimates of the system is empirically obtained for constrained- and free-motion cases by DSP Technology Inc.’s Siglab hardware [47]. Using

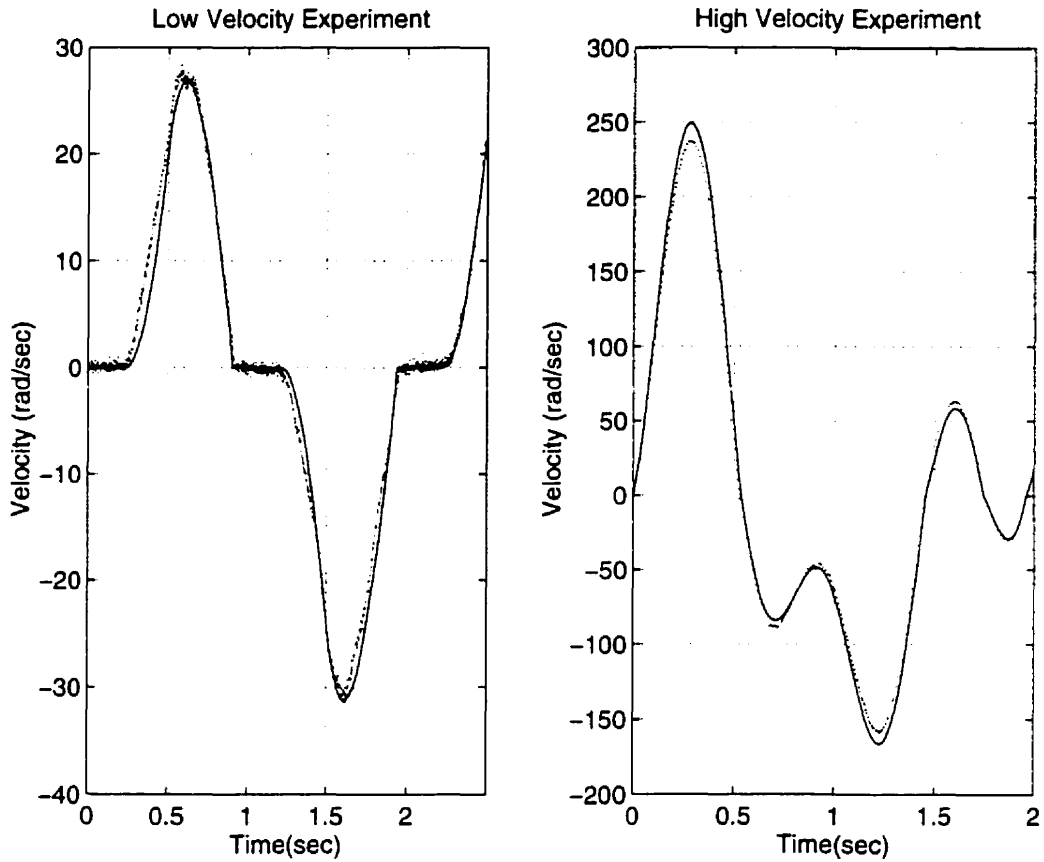


FIGURE 4.1. Velocity comparison of the model and experiment for two typical experiments; Solid: simulations, dotted: measured velocity

sine-sweep and random inputs with different amplitudes on each experimental setup, a set of frequency response estimates for the system is generated. Applying an iterative Gauss-Newton routine on one of the frequency response estimates, a transfer function is obtained which minimizes the weighted least-squares error between the experimental frequency response and the model¹. The McGill setup servo amplifier frequency response is illustrated in Figure 4.2, which is flat up to 300 *rad/sec*, and whose associated transfer function can be approximated by:

$$\frac{\text{Current}}{\text{Ref Voltage}} = \frac{5.617 \times 10^8}{s^3 + 2987s^2 + 4.66 \times 10^6s + 2.156 \times 10^9} \quad (4.5)$$

which has three stable poles at -707.67 , and $-1139.5 \pm j1322.2$ and a DC-gain of -11.68 dB. Similarly, ISE servo amplifier has the following transfer function:

¹Function *invfreqs* in Matlab

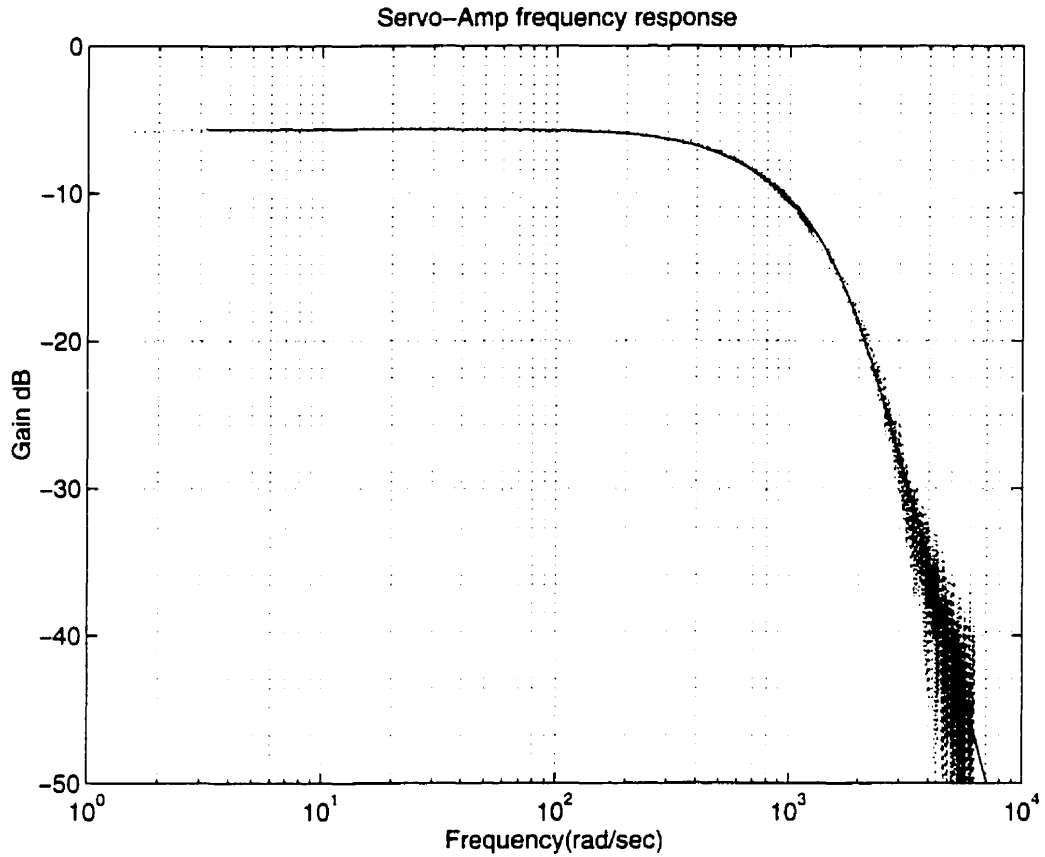


FIGURE 4.2. Frequency response of the servo amplifier

$$\frac{\text{Current}}{\text{Ref Voltage}} = \frac{2.48 \times 10^9 (s^2 - 5188s + 1.59 \times 10^7)}{s^5 + 4713s^4 + 2.34 \times 10^7 s^3 + 5.04 \times 10^{10} s^2 + 7.67 \times 10^{13} s + 3.72 \times 10^{16}} \quad (4.6)$$

where in this case the best estimate is a non-minimum phase system with zeros at $2593.9 \pm j3032.6$, poles at -732 , $-856.1 \pm j3437.0$, and $-1134.4 \pm j1661.3$, and a DC-gain of 0.55 dB. Moreover, the saturation limit for McGill servo amplifier is 3 *amps*, while it is 10 *amps* for ISE setup, as explained in Sections 1.1 and 1.2 of Chapter 2.

3. Harmonic Drive

A harmonic drive can be viewed as a two-input, two-output black box, where the inputs are the wave generator angular position θ_{wg} and the torque transmitted through the flexspline T_{fs} , while the outputs are the flexspline angular displacement θ_{fs} , and the



FIGURE 4.3. Harmonic drive system modelled as a black box

required torque at the wave generator T_{wg} as illustrated in Figure 4.3. An ideal model of the system would only describe the overall gear reduction property of the transmission, in absence of any frictional losses and compliance effect. This can be represented by:

$$\theta_{wg} = N \theta_{fs} \quad (4.7)$$

$$T_{wg} = \frac{1}{N} T_{fs} \quad (4.8)$$

where N is the gear ratio of the harmonic drive.

A complete model of the system, however, includes the frictional losses and compliance effect in addition to the reduction property of the transmission. Figure 4.4 represents the complete model of the system, in which the friction losses are separated into the wave generator bearing friction T_{f1} , gear meshing friction T_{f2} , output bearing friction T_{f3} and the flexspline structural damping T_{st} , while T_k represents the stiffness of the flexspline. The position and torque balance between each two nodes can be represented by the following equations:

$$\begin{cases} \theta_{wg} = \theta_{nwg} \\ T_{wg} = T_{nwg} + T_{f1} \end{cases} \quad (4.9)$$

$$\begin{cases} \theta_{nwg} = N \cdot \theta_{nfs} \\ T_{nwg} = \frac{1}{N} T_{nfs} + T_{f2} \end{cases} \quad (4.10)$$

$$\begin{cases} T_{nfs} = T_{cfs} \\ T_{nfs} = T_k + T_{st} \end{cases} \quad (4.11)$$

$$\begin{cases} \theta_{cfs} = \theta_{fs} \\ T_{fs} = T_{cfs} - T_{f2} \end{cases} \quad (4.12)$$

The compliance model relates the stiffness T_k and structural damping T_{st} to the relative torsion of the flexspline $\Delta\theta = \theta_{nfs} - \theta_{cfs}$. The following Sections 3.1, and 3.2 elaborate our proposed model for compliance, and friction respectively.

3.1. Harmonic Drive Compliance

As described in the manufacturer's catalogue [60], a typical shape of the harmonic drive compliance curve is as the experimental result of Figure 4.5. This curve illustrates harmonic

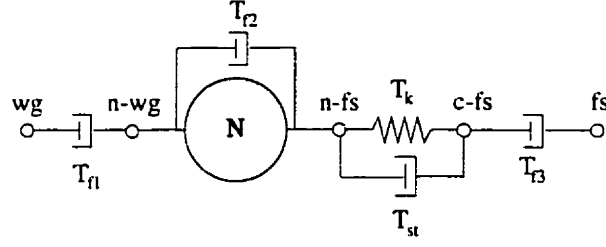


FIGURE 4.4. Transmission model of the harmonic drive with compliance and friction

drive nonlinear stiffness and hysteresis. To capture the nonlinear stiffness behavior, manufacturers suggest using piecewise-linear approximations [60], whereas other researchers prefer a cubic polynomial approximation [121, 124]. The hysteresis effect, however, is more difficult to model, and consequently it is often ignored. Recently Seyfferth et al. proposed a fairly complex model to capture the hysteresis [97]. The hysteresis in the harmonic drive compliance profile is caused by structural damping of the flexspline. The inherent coupling of stiffness torque and structural damping, therefore, makes it very hard to identify those separately.

We suggest that Figure 4.5 is in fact a Lissajous figure, and that we identify both the stiffness and damping of the flexspline together using least-squares estimation. Linear and cubic models for compliance and many different models for structural damping were tried in this framework. The Dahl model for friction [29, 116] and the Duham, Preisach and Babuska models for hysteresis [81], are among the many dynamic models used to replicate the hysteresis torsion curve. The details of these dynamic models and their identification results are elaborated in [102]. We observed, however, that a linear stiffness model accompanied with a static model which relates the structural damping to a power of the velocity, can best capture hysteresis behavior. The reason why dynamic models were not capable of accurately predicting hysteresis in harmonic drive structural damping is that despite their dynamic relation, the dependence of the structural damping torque to a power of the velocity was not accommodated. Hence, our proposed model, simpler in structure, appears to better characterize the hysteresis. In practice the consistency measure of identified parameters in our model is much less than those of other dynamic models we examined.

Equation 4.13 gives in detail the compliance model, where $\Delta\theta$ is the flexspline relative torsion.

$$T_{meas} = K_1 \Delta\theta + T_{st} |\Delta\dot{\theta}|^\alpha \text{sign}(\Delta\dot{\theta}) \quad (4.13)$$

To identify the model parameters, a set of constrained motion experiments has been em-

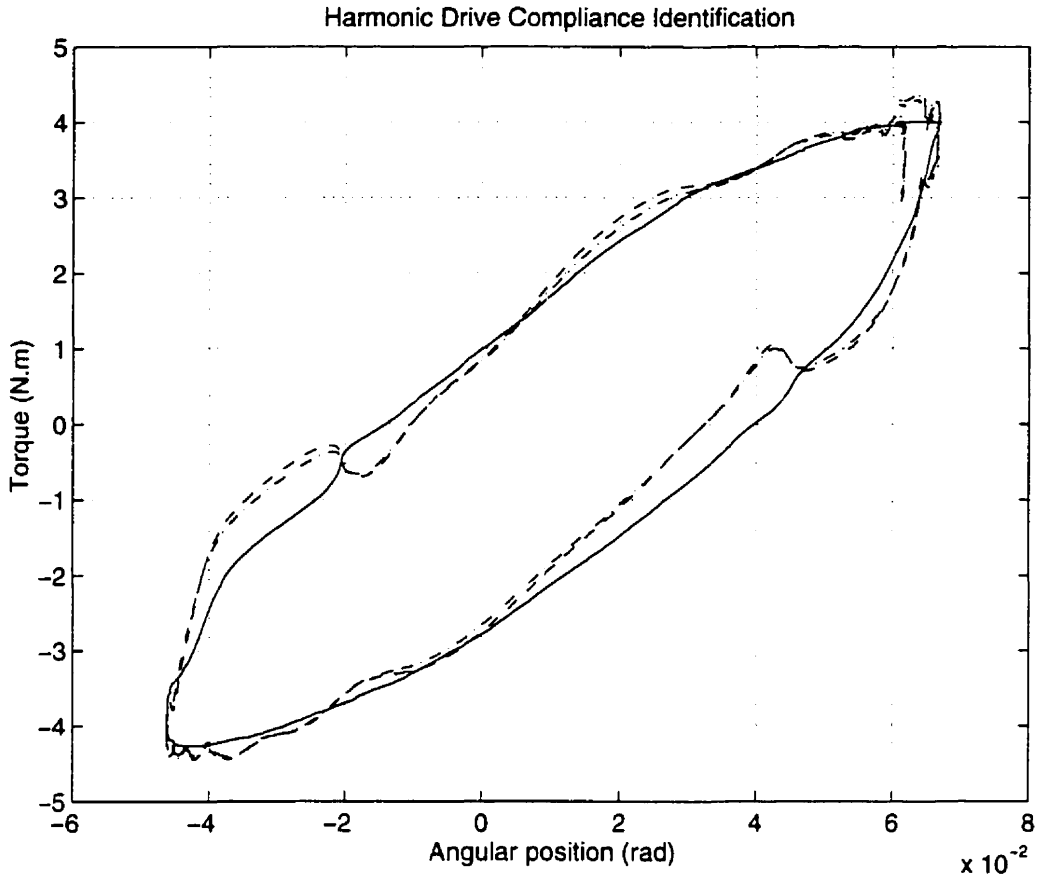


FIGURE 4.5. Typical hysteresis curve and its model for optimal α (dash-dot), and $\alpha = \frac{1}{2}$ (dashed)

ployed, in which the torque T_{meas} and the motor velocity have been measured. The experiment shape functions are the same as that explained in DC motor experiments. Equation 4.13 forms a nonlinear regression in which K_1 , T_{st} and α are unknown. Using an iterative least-squares solution for this nonlinear regression model, it is found that the optimal estimate of α is very close to 0.5. Consequently the structural damping can be related to the square root of the relative torsion velocity. Figure 4.5 illustrates a typical hysteresis torsion curve fitted by the model, comparing the difference between the optimal α and $\alpha = \frac{1}{2}$. The maximum mismatch (points $(-2, -0.5)$ and $(0.5, 0.8)$ in Figure 4.5) occurs when the velocity is changing rapidly; otherwise, the model is approximating the hysteresis curve quite accurately. By fixing the value of $\alpha = \frac{1}{2}$, Equation 4.13 forms a linear regression model for the system and can be solved for different experiments. Table 4.2 summarizes the compliance parameter for the harmonic drives of our two setups.

TABLE 4.2. Harmonic drives estimated parameters

	McGill Harmonic Drive		ISE Harmonic Drive	
	Estimated Parameter	Consistency Measure	Estimated Parameter	Consistency Measure
α	$\frac{1}{2}$	0%	$\frac{1}{2}$	0%
K_1	6340	9.6%	104.2	4.36%
T_{st}	57.2	28.2%	7.96	28.0%
J_{eff}	1.0×10^{-4}	6.38%	1.0×10^{-4}	8.72%
T_{vp}	3.7×10^{-4}	16.7%	1.8×10^{-3}	13.2%
T_{vn}	3.5×10^{-4}	19.3%	2.1×10^{-3}	8.42%
T_{sp}	4.6×10^{-2}	23.7%	7.5×10^{-2}	29.2%
T_{sn}	4.4×10^{-2}	24.0%	3.3×10^{-2}	30.8%
T_{sspl}	-0.0076	14.7%	-0.0487	20.3%
T_{ssn1}	-0.0203	23.8%	-0.0450	18.6%
T_{ss2}	0.1	0%	0.1	0%

3.2. Harmonic Drive Friction

All harmonic drives exhibit power loss during operation due to transmission friction. Figure 4.4 illustrates the schematics of the harmonic drive model. The bulk of energy dissipation can be blamed on the wave generator bearing friction T_{f1} , gear meshing friction T_{f2} , output bearing friction T_{f3} and the flexspline structural damping T_{st} . Among them, most of the frictional dissipation results from gear meshing. Also comparing the ball-bearing frictions, T_{f1} is more important than T_{f3} since it is acting on the high speed/low torque port of transmission, and its effect on the dynamics of the system is magnified by the gear ratio. The transmission torque is measured directly by strain gauges mounted on the flexspline (namely node c_{fs} of Figure 4.4). The torque balance, therefore, can be written as:

$$T_{wg} = \frac{1}{N}(T_{meas}) + T_{f1} + T_{f2} \quad (4.14)$$

in which the measured torque $T_{meas} = T_k + T_{st}$, N is the gear ratio, and T_{wg} is the resulting torque of the wave generator, provided by the DC motor. From Equation 4.1, T_{wg} can be related to the input current by:

$$T_{wg} = K_m i - J_m \ddot{\theta} - T_{fm} \quad (4.15)$$

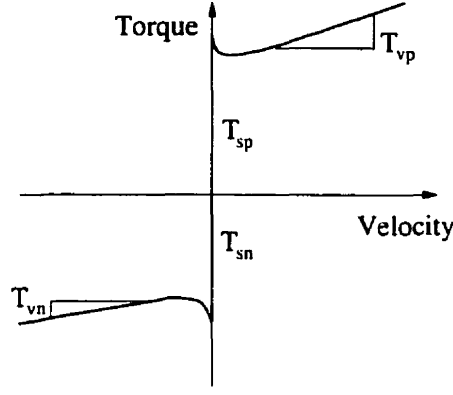


FIGURE 4.6. Friction torque modelled as a superposition of Coulomb, viscous and Stribeck friction, having velocity-direction-dependent coefficient

Thus, the final torque balance of the system is the following:

$$K_m i - \frac{1}{N} T_{meas} = J_{eff} \ddot{\theta}_{wg} + (T_{fm} + T_{f1} + T_{f2}) \quad (4.16)$$

in which K_m is the motor torque constant, J_{eff} is the effective input inertia, and T_{fm} is the motor friction. The gear meshing friction torque is modelled as Coulomb, viscous and Stribeck friction [9, 117], having velocity-direction-dependent coefficient as illustrated in Figure 4.6. This can be represented by:

$$\begin{aligned} T_{f2} = & T_{vp} \dot{\theta}_{wg} u_{-1}(\dot{\theta}_{wg}) + T_{vn} \dot{\theta}_{wg} u_{-1}(-\dot{\theta}_{wg}) + \\ & T_{sp} \text{sign}(\dot{\theta}_{wg}) u_{-1}(\dot{\theta}_{wg}) + T_{sn} \text{sign}(\dot{\theta}_{wg}) u_{-1}(-\dot{\theta}_{wg}) + \\ & T_{ssp1} \text{sign}(\dot{\theta}_{wg}) u_{-1}(\dot{\theta}_{wg}) e^{-\left(\frac{\dot{\theta}_{wg}}{T_{ssp2}}\right)^2} + T_{ssn1} \text{sign}(\dot{\theta}_{wg}) u_{-1}(-\dot{\theta}_{wg}) e^{-\left(\frac{\dot{\theta}_{wg}}{T_{ssn2}}\right)^2} \end{aligned} \quad (4.17)$$

where the function u_{-1} is defined in Equation 4.3.

The Stribeck model for friction can capture the dynamics of the friction at low velocities. Unlike compliance identification, both constrained and free motion experiments are employed to identify the friction model parameters. Free-motion experiments are suitable for viscous and Coulomb friction, while constrained-motion experiments operate the system at low velocities which are ideal for Stribeck coefficient identification. Free motion low-velocity experiments are used as well, for Stribeck coefficient identification. The experiment inputs are the same as that explained in DC motor experiments, where for constrained motion case 20 experiments, and for free motion case 30 experiments are considered for each setup.

Equation 4.16 forms a linear regression model for the high velocity experiments in the absence of the nonlinear Stribeck terms. Viscous and Coulomb friction coefficients T_{vp} , T_{vn} , T_{sp} and T_{sn} , and the inertia J_{eff} is obtained from least-Squares solution to this

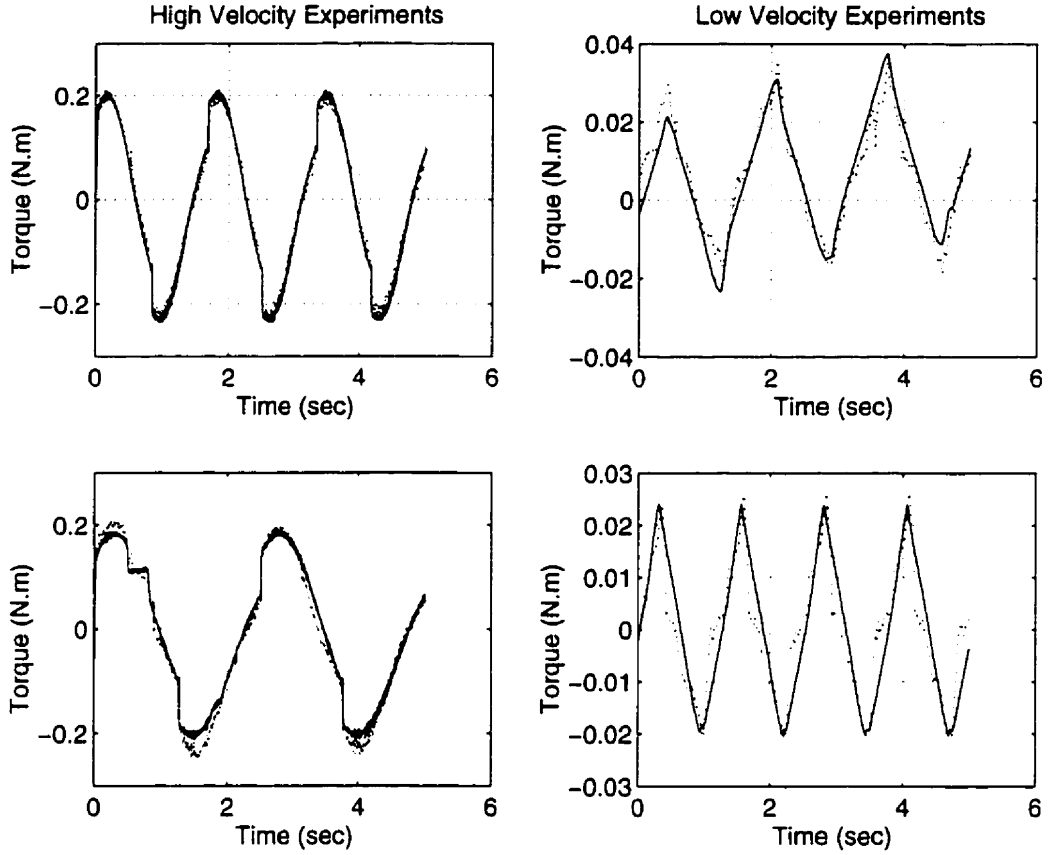


FIGURE 4.7. Comparison of output torque in simulations and experiments; dotted: experiment, solid: simulation

case. For low-velocity experiment also, Equation 4.16 can provide a linear regression if $T_{ss2} = T_{ssp2} = T_{ssn2}$ is assumed to be known. Table 4.2 summarizes the estimated friction parameters of two harmonic drives, and their consistency measure and Figure 4.7 illustrates the output torque fit obtained by the model for four typical experiments, assuming fixed $T_{ss2} = 0.1$. The consistency measure for all parameters are less than 30%, which indicates the reliability of the estimated parameters. It should be noted that in this regression model instead of the internal system friction T_{f_m}, T_{f_1} and T_{f_2} , the entire friction of the system ($T_f = T_{f_m} + T_{f_1} + T_{f_2}$), can be identified. This imposes no limitation on the identification procedure, since only the entire friction T_f is required for the simulations.

It is important to note that the estimated Stribeck friction coefficients are negative, which is in contrary to the usual dynamics of friction reported at low velocities [8, 55]. Nevertheless, this represents rising friction at low velocities as illustrated in Figure 4.8, and no stiction, verifying the manufacturers claim [22]. This may be rationalized by the fact

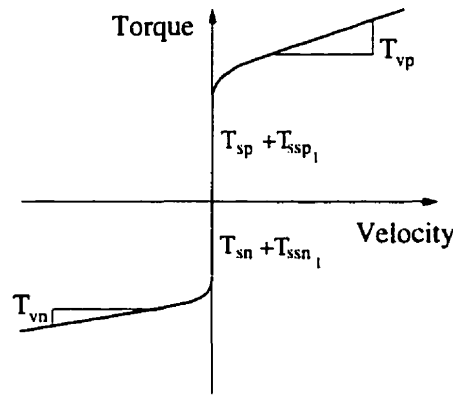


FIGURE 4.8. Identified friction curve

that the main bulk of frictional losses in the harmonic drive systems are due to the gear meshing but that, contrary to other transmissions, a combination of elastic deformation of the flexspline and gear teeth engagement contributes to the velocity reduction. Therefore, the low-velocity experiments in the harmonic drive transmission shows smoother startup velocity when compared to other transmissions. This is verified by both constrained and unconstrained motion experiments, where no stick-slip or stiction is observed. The reliability of the negative Stribeck coefficient is assessed first by the acceptable consistency measure for the Stribeck coefficients, and second by the similar results for the two different harmonic drives.

4. Summary

Based on experimental and theoretical studies, a systematic way to capture and rationalize the dynamics of the harmonic drive systems is introduced. Simple and accurate models for compliance, hysteresis, and friction are established and model parameters are identified using least-squares approximation. A measure of consistency is defined, by which the reliability of the estimated parameter for different operating condition, as well as the accuracy of the simple model is quantified. From compliance modelling results, it has been shown that identifying stiffness and structural damping together will resolve the reported difficulties in determining the compliance parameters. Moreover, it has been shown that a linear stiffness model best captures the behavior of system when combined with a good model for hysteresis. A simple static model for hysteresis is also introduced, and it is shown that this simple model can replicate the hysteresis effect in harmonic drives better than

some other more complex dynamic models reported in the literature. Friction losses of the harmonic drive are modelled at both low and high velocities. From experiments on two different harmonic drives it is observed that there is no stiction in the transmission, but rather a rising friction acts at low velocities. Finally, the model performance is assessed by a simulation verifying the experimental results for both constrained- and free-motion systems.

CHAPTER 5

Simulations

In Chapter 4 a methodology for accurate modelling of harmonic drive system components was proposed. Computer simulations can assist in verifying the integrity of the whole model and evaluating the accuracy of the identified parameters. In this chapter we address the details of simulations designed for model verification, in addition to the general simulation of the system for constrained and free motion cases. Finally, the significance of the nonlinear model is assessed through a parameter sensitivity study using simulations.

1. Modelling and Identification Verification

To verify the validity of the modelling scheme, simulations of the system under constrained and free-motion cases are developed in the Simulink. The equation of motion for each component of the system is integrated into a simulation unit, and the simulation is initialized by downloading the identified parameters. In order to compare the simulation result of the integrated system to the experiments, an experiment data file is also downloaded into the simulation environment. Hence, using measured DC motor current of a typical experiment as an input to the simulation, the velocity and torque output of the simulated system are compared to that of the experiment. By this means the ability of the simulation to predict the dynamic behaviour of the system is studied.

1.1. System under Constrained-Motion

Figure 5.1 displays the simulation unit of the constrained-motion system. The internal simulation blocks of the DC motor and the harmonic drive is also illustrated in Figure 5.1. The DC motor simulation block consists of a double-integrator corresponding to the Equation 4.1. The current input is clamped with a saturation block, and the friction torque simulates Equation 4.2. The harmonic drive simulation block simulates Equations 4.9 to

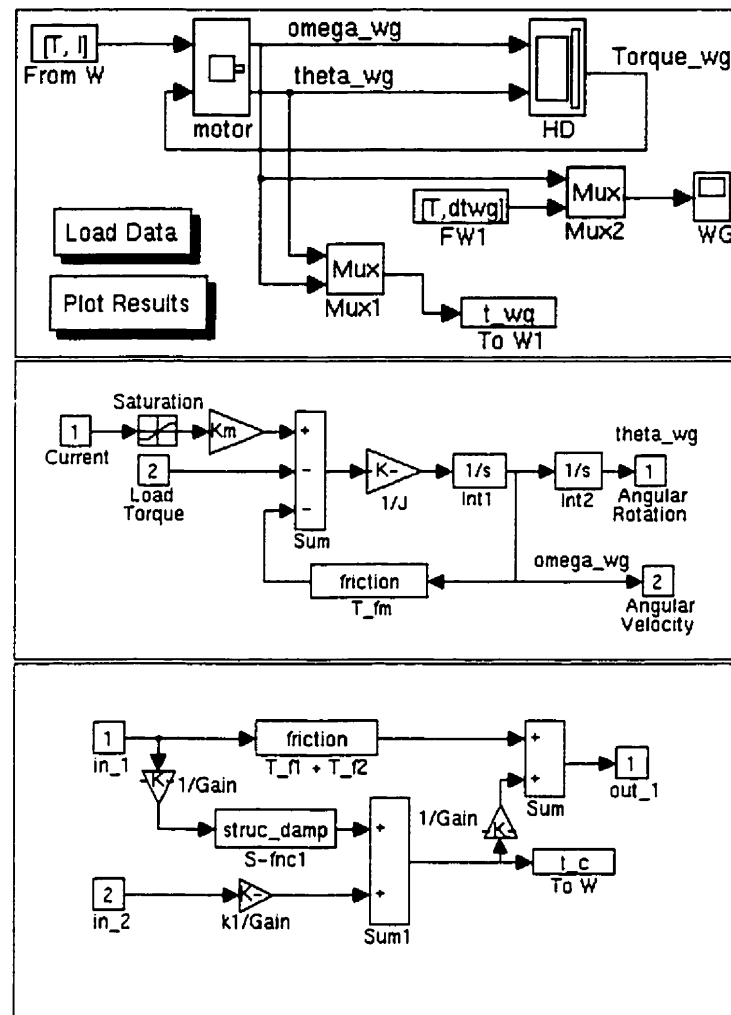


FIGURE 5.1. Harmonic drive system under constrained-motion simulated on Simulink for model verification purpose. Top: System; Mid: DC-motor; Bot: Harmonic drive

4.12, in which the flexspline position is set to zero, consistent with the constrained-motion requirement. The structural damping and friction blocks simulate Equations 4.13 and 4.17 respectively.

Special attention is given to simulate friction, because of its inherent hard nonlinearity. Figure 4.6 illustrates the discontinuity of the friction function at zero velocity. It is inappropriate to implement this model immediately by computer, since absolute zero velocity is infeasible due to the round-off errors. To overcome this problem a small threshold is usually considered instead of absolute zero. Because of the discontinuity in the friction function, however, introducing a threshold may result in chattering velocity. Haessig and Karnopp

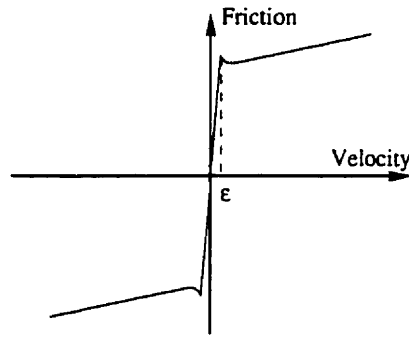


FIGURE 5.2. The modified friction function used for simulations to avoid chattering.

proposed smoothing the friction function near zero velocity to overcome this chattering problem [48, 70]. Figure 5.2 illustrates the replacement of the discontinuity by a curve of finite slope within a small boundary layer near zero velocity. The boundary layer width ϵ significantly attenuates the chattering while at the same time increasing the simulation speed. We proposed a variable width boundary layer whose size will be dynamically determined in the simulation code as a function of the integration time-step, system moment of inertia, driving torque, and identified friction parameters. The details of this implementation are elaborated in [102], where the significance of this implementation step is shown through a simulation study. It is observed that the boundary layer width ϵ remains always smaller than 10^{-4} (rad/sec) which is much smaller than 0.1 (rad/sec), the velocity region where Stribeck friction is dominant. Hence, this modification has little effect on Stribeck friction implementation in the simulations.

In order to start simulations, as a user-friendly feature, it is sufficient to double click on the “Load Data” block in the main simulation unit. By this means, the model parameters are downloaded into the Matlab workspace, as well as the experimental data file. By pressing the “Plot Results” block after simulation termination, the final velocity and torque comparison plots are automatically generated in the Matlab environment. A Runge-Kutta fixed-step integration method with 1ms integration step is used to execute these simulations, consistent with the 1 kHz sampling frequency in the experiments. Furthermore, in order to accelerate the simulations, all internal nonlinear functions (*i.g.* friction block) are written in C as Simulink S-functions and compiled using the Cmex compiler. The discussion of the simulation results for both constrained- and free-motion cases is presented in § 1.3.

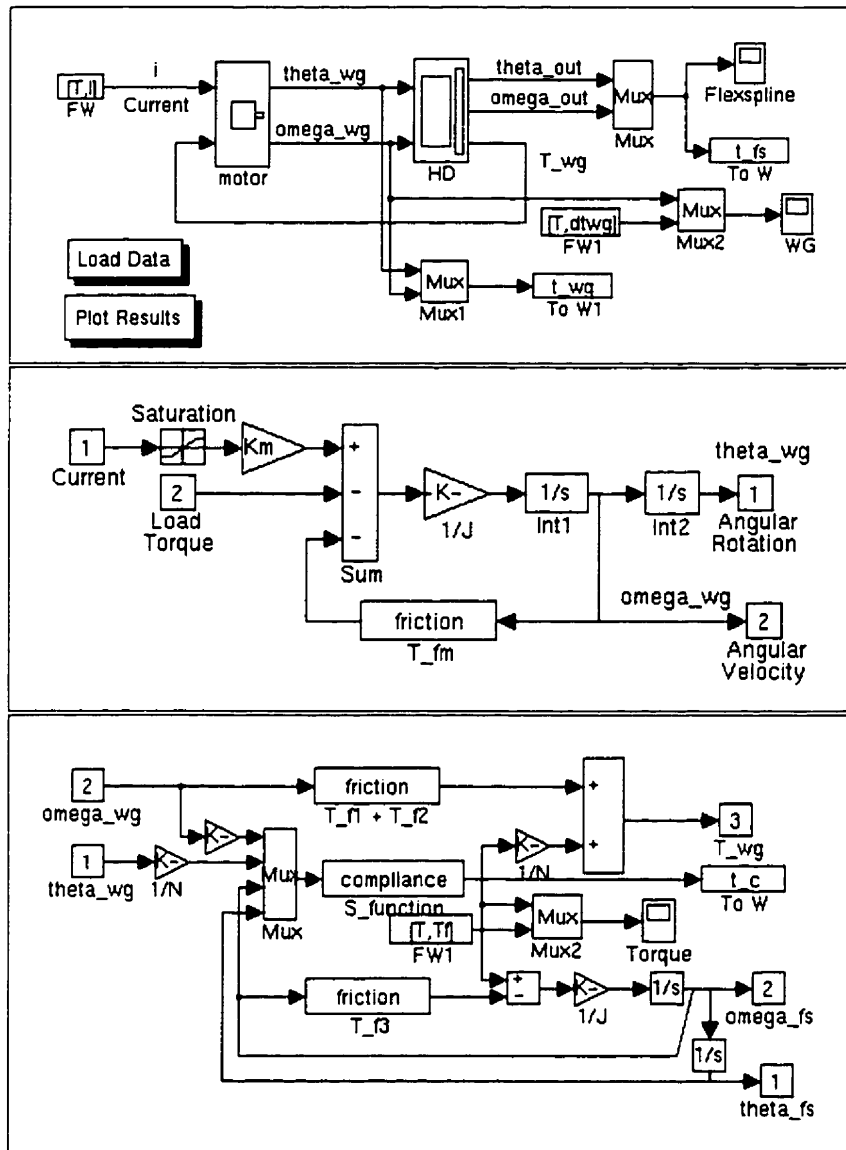


FIGURE 5.3. Harmonic drive system under free motion simulated on Simulink for model verification purpose. Top: System; Mid: DC-motor; Bot: Harmonic drive

1.2. System under Free-Motion

Similarly, Figure 5.3 illustrates a simulation unit for the free-motion system. The internal simulation blocks of DC motor and harmonic drive are shown separately. The DC motor simulation block have the same structure as in Figure 5.1, while the harmonic drive block is the simulation of Equations 4.9 to 4.12 in their general form. The compliance block also simulates Equations 4.13, in its general form where there is relative torsion in the flexspline structure. Similar to Figure 5.1, measured current in the experiments are used as

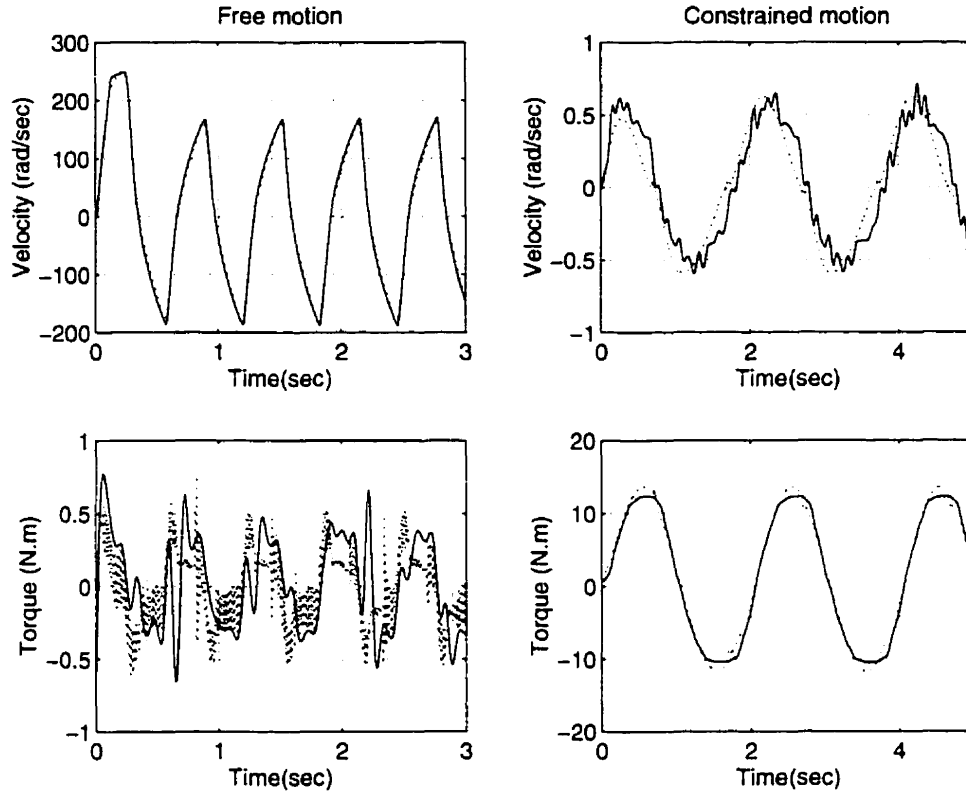


FIGURE 5.4. Simulation verification for system under free and constrained-motion; Solid : Experiment, Dotted : Simulation

an input to the simulations, while the simulated torque and velocity of the system can be compared to that of the experiments. The velocity and torque comparison of system under both constrained and free motion with experimental results is detailed in the next section.

1.3. Discussions

The output velocity and torque of the simulated system are compared to typical experimental outputs in Figure 5.4. First the experimental signals are filtered using a fifth-order zero-phase distortion Butterworth filter¹. Therefore, the torque ripples in free motion experiments, and the noise on the measured velocity in constrained-motion experiments are not displayed in Figure 5.4. For the system under free-motion, there is an almost perfect match for the velocity, and a relatively good match for the torque curves. The perfect match between the simulation and experiment velocity indicates the ability of the simulation to predict the dynamic behavior of the system. Some oscillations are observed in the simulated torque of the system which are absent in the filtered experimental torque displayed

¹Function *filtfilt* in Matlab

in Figure 5.4. The measured torque however, displays similar oscillation (torque ripples), but with larger amplitude. It should be mentioned that to have a model for accurate prediction of the torque ripples requires a complex gear meshing mechanism modelling, [121], which is not pursued in this research. Instead, in Chapter 3 a Kalman filter is employed to estimate the torque ripples, using simple harmonic oscillator model. Despite the simple model used for the prediction, it is shown that Kalman filter can accurately estimate the torque ripples [110].

For the system under constrained motion, the match between velocities is less accurate compared to that for the free motion system, because of the smaller velocity signal and hence smaller signal-to-noise ratio. However, the resulting torques are quite similar and there is no torque ripple observed for the constrained system. This accurate match was verified for more than twenty disparate experiments for the McGill and the ISE harmonic drive testing stations. Extensive results are provided for both constrained- and free-motion experiments in Appendix C. The accurate match between simulation and experiment for different operating ranges indicates the fidelity of the model to accurately predict the dynamic behavior of the system and confirms the effectiveness of modelling and parameter identification schemes to capture the dynamics of the harmonic drive systems.

2. System Simulations

2.1. System under Constrained-Motion

A separate simulation module for the system under constrained motion is developed in Simulink and is illustrated in Figure 5.5. This module could be incorporated into a feedback algorithm, so as to evaluate the performance of the closed loop system by simulations. The reference input to the system is produced by a signal generator, and the servo amplifier

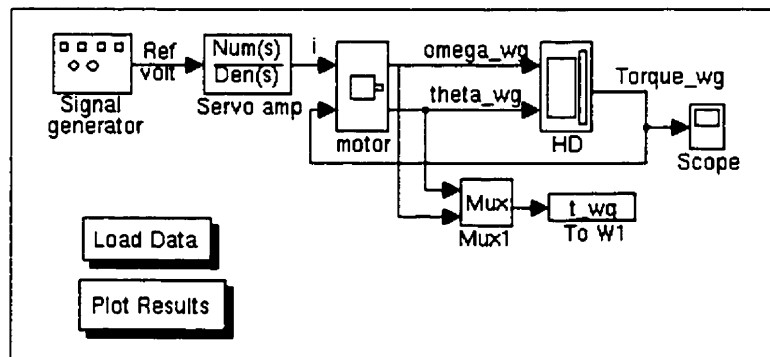


FIGURE 5.5. Harmonic drive system under constrained-motion simulated on Simulink

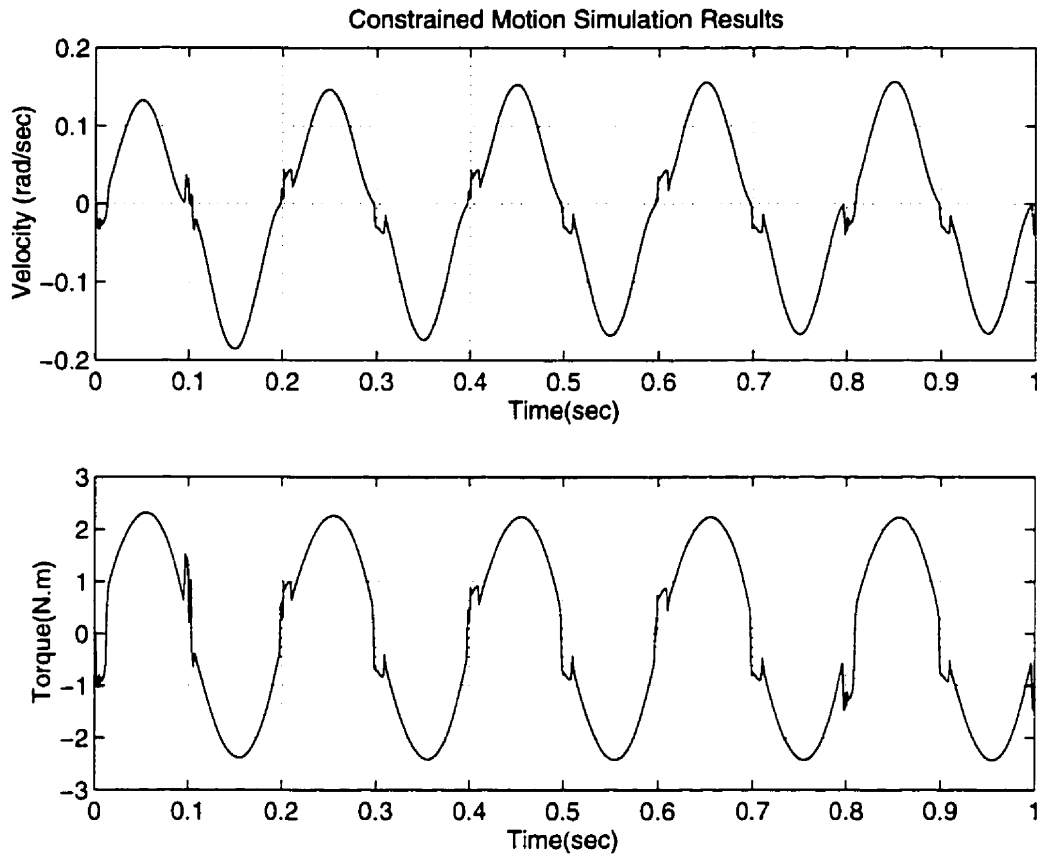


FIGURE 5.6. A typical simulation result of the harmonic drive system under constrained-motion

transfer function identified in Chapter 4 is added into the simulation block. The DC motor and harmonic drive simulation blocks however, are identical to that on Figure 5.1. Variable-step Runge-Kutta method is used for integration. A typical simulation result is illustrated in Figure 5.6 for a sinusoid input. The output velocity of the system displays different amplitudes in positive and negative velocity regions. This is due to the dependence of the friction coefficients to the velocity direction, as accurately represented by Equation 4.17. The torque output displays a near-discontinuous shape which is caused by the discontinuity of the Coulomb friction. The phase lag of the velocity and torque signals which occurs twice in each cycle is due to the hysteresis embedded in the structural damping representation of the harmonic drive.

2.2. System under Free-Motion

Similar to the constrained-motion simulation module, a separate simulation module for free-motion system is developed in Simulink as in Figure 5.7. A typical simulation result

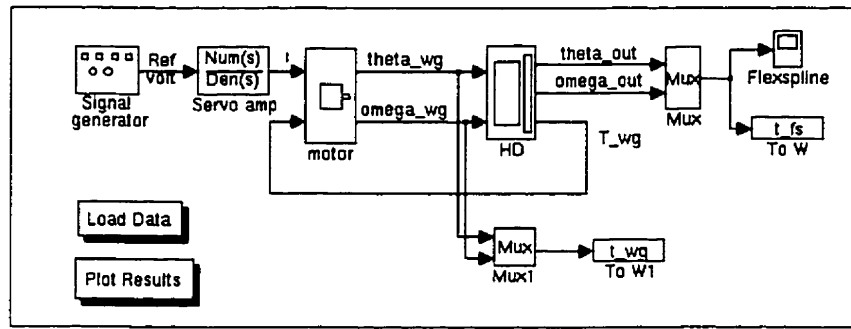


FIGURE 5.7. Harmonic drive system under free-motion simulated on Simulink

for sinusoid input is illustrated in Figure 5.8, where the output torque, wave generator velocity and flexspline velocity are displayed. Variable-step Runge-Kutta method is used for integration. Except a small interruption in the smooth trend of the torque output of the system, no torque oscillation is observed in this simulation. The wave generator and flexspline velocity output of the system are also very smooth with a small delay due to the

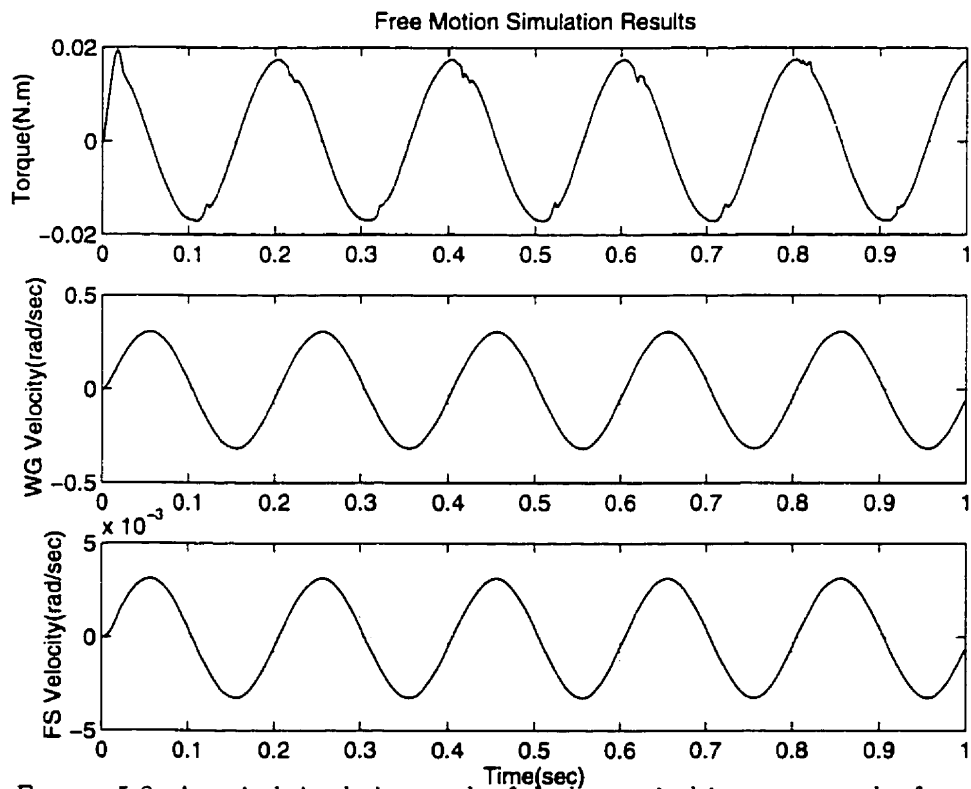


FIGURE 5.8. A typical simulation result of the harmonic drive system under free-motion for sinusoid input. Top: Torque, Mid: Wave generator velocity, Bot: Flexspline velocity

hysteresis effect of the structural damping representation. This module could likewise be further enhanced to incorporate a torque feedback algorithm.

3. Parameter Sensitivity Study

The significance of individual components of the nonlinear model is assessed by a parameter sensitivity study using simulations. In this study, the simulation results of the complete model and a series of simplified models are compared to the experiments. To examine the effect of Coulomb and viscous friction, the free-motion simulation of the system, constructed for model verification (as in § 1.2 and Figure 5.3), is employed. For a simplified model, either the Coulomb or the viscous friction coefficient is set to zero, and the simulation results of the complete and simplified model are compared to the experiment. Figure 5.9 illustrates the comparison results for a typical experiment. In absence

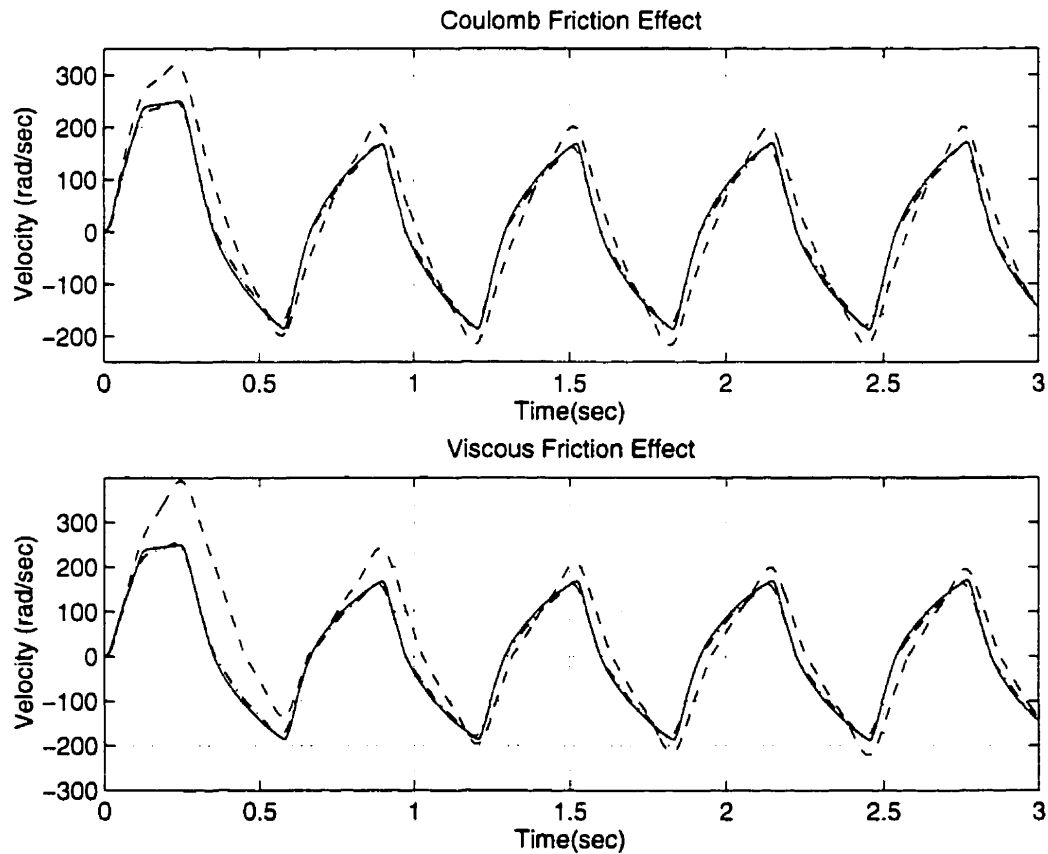


FIGURE 5.9. Comparison of the experiment with the complete and simplified models, where the significance of Coulomb and viscous friction are examined separately; Solid : Experiment, Dash-dotted : Complete model, Dashed : Simplified model

of either Coulomb or viscous friction the model is not capable of accurately estimating the experimental result. Hence, we may conclude that accurate identification of Coulomb and viscous friction coefficients significantly contributes to the accuracy of the simulations. Stribeck friction, however, is only present in very low-velocity experiments, and its effect in the typical experiment of Figure 5.9 is negligible. However, in applications where fine motions at low velocities are demanded, modelling and identification of Stribeck friction may be prominent.

To examine the effect of the compliance model the constrained-motion simulation of the system, constructed for model verification (as in § 1.1 and Figure 5.1), is employed. For simplified models first the stiffness is reduced by one order of magnitude, and then the structural damping coefficient is set to zero, while keeping all the other identified parameters of the model. Figure 5.10 illustrates the comparison results of the complete and simplified

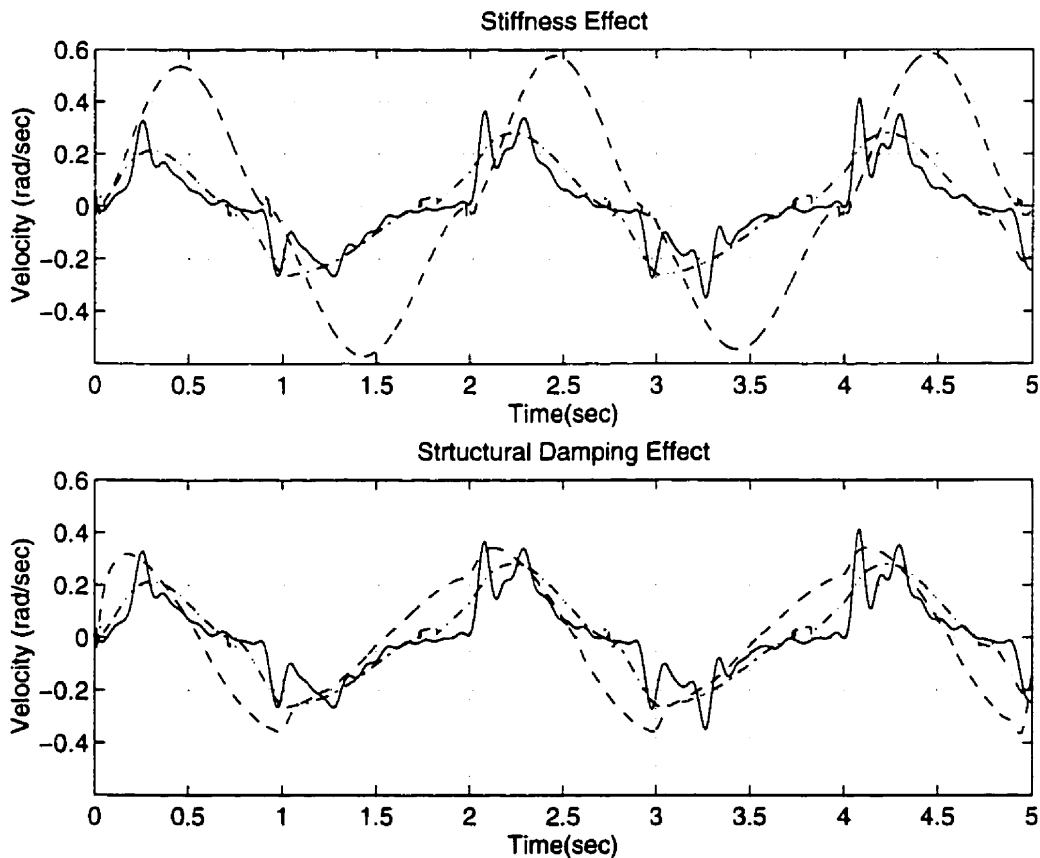


FIGURE 5.10. Comparison of the experiment with the complete and simplified models, where the significance of stiffness and structural damping are examined separately; Solid : Experiment, Dash-dotted : Complete model, Dashed : Simplified model

models to the experiment. It is observed in this figure that the lack of accuracy in identifying the stiffness parameter results in larger velocity amplitude in the simulations. Furthermore, an inaccurate structural damping coefficient results in phase estimation error of the velocity. Therefore, both stiffness and structural damping estimation contribute significantly into accuracy of the model.

4. Summary

The model performance is assessed by a simulation verifying the experimental results for both constrained- and free-motion cases. For the purpose of modeling scheme verification simulations of the system is developed in Simulink. The simulated velocity and torque of the system is compared to experimental results for several constrained- and free-motion experiments. An accurate match between simulation and experiment for different operating ranges are obtained, which indicates the fidelity of the model to accurately predict the dynamic behavior of the system. Moreover, it confirms the effectiveness of modelling and parameter identification schemes to capture the dynamics of the harmonic drive systems.

Furthermore, the significance of individual components of the nonlinear model is assessed by a parameter sensitivity study. By comparing the simulation results of the complete model and a series of simplified models to the experiment results, it is concluded that accurate estimation of Coulomb and viscous friction significantly contributes to the accuracy of the simulations. Finally, it is shown that both stiffness and structural damping are prominent for the accuracy of the constrained-motion simulations.

CHAPTER 6

Robust Torque Control

In numerous robotic control techniques such as feedback linearization, computed torque method and some adaptive control schemes, the actuator torque is taken to be the control input [89, 99, 100]. The physical variable being manipulated in practice, however, is not torque but armature current in a DC motor, for instance. For harmonic drive systems the relation between output torque and input current possesses nonlinear dynamics, due to the flexibility, Coulomb friction and structural damping of the harmonic drive [103]. Therefore, it is desired to improve this input/output relation by *torque feedback*, and to convert the system to a near-ideal torque source with a flat frequency response over a wide bandwidth.

There is a dichotomy in torque-control applications for a robot manipulator using harmonic drives in its joints. First are applications where the robot is in contact with a stiff environment and high torques at very low velocities are required at each joint. Simulation of this application at each joint can be studied by a constrained-motion experiment. The second class of applications occurs when the robot's links are moving freely, and the only torque required at each joint is to compensate for gravity, friction and link acceleration. This problem can be simulated through a free-motion model, especially when the gear ratio is large enough for the motor inertia to dominate. In the free-motion case the amount of torque required at each joint is very low but at much higher velocities.

In this chapter a general framework to design robust torque controllers for harmonic drive system is developed and tested for constrained-motion and free-motion experiments. It is shown that an empirical linear model of the system and an associated uncertainty representation are sufficient to build a robust torque controller. An $\mathcal{H}_\infty$ -framework is used for controller design, and the proposed controller is tested on the McGill, and the ISE setups. Exceptional performance results are obtained from the time and frequency response

of the closed-loop system, especially for constrained-motion application. To further improve the performance of the system for the free-motion case, a friction-compensation algorithm is implemented on the system. It is shown that by this method the frequency response variation of the system is reduced significantly. By comparison of the frequency domain performance of the system with and without friction-compensation, it is shown that friction-compensation will improve the performance of the system for tracking signals with low-frequency content.

1. System Linear Model and its Uncertainty

A complete model of the system was derived in Chapter 4. To capture the system dynamics accurately, it is shown that it is necessary to consider nonlinear models for friction and structural damping [103]. For the purpose of control however, a linear model for the system is required for the controller design synthesis. We propose an empirical method to find this nominal model by performing a series of frequency response experiments on the system with different input amplitudes, and to find the best fit through them. In this method, not only the empirical nominal model of the system (without need of any linearization) will be determined, but also variations in the frequency response of the system, due to the nonlinearities, will be encapsulated with an uncertainty representation.

As described in Chapter 2, DSP Technology Inc. Siglab hardware is used to generate the empirical frequency response estimates of the system [47]. Using Siglab-generated sine-sweep and random noise inputs with different amplitudes on each experimental setup, an ensemble of frequency response estimates for the system is generated. Applying an iterative Gauss-Newton routine on one of the frequency response estimates, a transfer function is obtained which minimizes the weighted least-squares error between the experimental frequency response and the model¹. We call this transfer function the “Nominal Model” of the system (illustrated in Figures 6.1 and 6.2). Moreover, the variation of each frequency response estimate from the nominal model can be encapsulated by a multiplicative uncertainty. Assuming that the nominal plant transfer function is $\mathbf{P}_0(s)$, we define $\mathcal{P}$ as the family of possible models of the system including all the experimental frequency response estimates, and the nominal model of the system. As elaborated in § 2.3.1 of Appendix A, by multiplicative uncertainty we mean:

$$\forall \mathbf{P}(s) \in \mathcal{P}, \quad \mathbf{P}(s) = (\mathbf{1} + \Delta(s)\mathbf{W}(s))\mathbf{P}_0(s). \quad (6.1)$$

¹Function *invfreqs* in Matlab

Here $\mathbf{W}(s)$ is a fixed transfer function, called the *uncertainty weighting function* and Δ is a variable $\mathcal{H}_\infty$ transfer function of norm less than unity [36]. Note that in this representation $\Delta(s)\mathbf{W}(s)$ gives the normalized system variation away from 1 at each frequency:

$$\frac{\mathbf{P}(j\omega)}{\mathbf{P}_0(j\omega)} - 1 = \Delta(j\omega)\mathbf{W}(j\omega) \quad (6.2)$$

Now as $\|\Delta\|_\infty \leq 1$, we may conclude that

$$\left| \frac{\mathbf{P}(j\omega)}{\mathbf{P}_0(j\omega)} - 1 \right| \leq |\mathbf{W}(j\omega)|, \quad \forall \omega \quad (6.3)$$

By plotting the system variations $\left| \frac{\mathbf{P}(j\omega)}{\mathbf{P}_0(j\omega)} - 1 \right|$ for all experimental frequency response estimates of the system $\mathbf{P}(j\omega)$, and estimate an upper bound as a transfer function, a multiplicative uncertainty weighting function $\mathbf{W}(s)$ can be obtained (as illustrated in Figures 6.1 and 6.2).

As another method to obtain a linear model for the system, the nonlinear equations of motion of the system, given in Chapter 4, can be linearized in a neighbourhood of the origin of the state space, which can be shown to be an equilibrium point for the system. The linear model derived by this means is called as “Theoretical Model” and illustrated in Figure 6.1. The main difference between the nominal model and the theoretical model is the latter’s inclusion of a resonance, mainly due to ignoring the nonlinear friction terms in the linearization process of the theoretical model. For the purpose of control synthesis, the nominal model of the system gives better representation of the true dynamics and thus is used for controller design. Note that by this method an effective way to find the closest linear model for a nonlinear system is proposed, and the deviation of the nonlinear system and linear model is encapsulated in model uncertainty. For our harmonic drive system the model uncertainty at low frequencies, (as illustrated in Figures 6.1 and 6.2), is relatively small and about -5 db, which suggest the possibility of robustly controlling the system to perform within this bandwidth.

1.1. System under Constrained-Motion

The methodology elaborated in § 1 is applied to the McGill and ISE setups to obtain a nominal model and uncertainty. Since the results are similar, here we report only the results of former setup, while the details of the latter can be found in [107]. Figure 6.1 illustrates the empirical frequency response of the McGill setup under constrained-motion, including nominal model and its uncertainty. The nominal model for this setup is found to

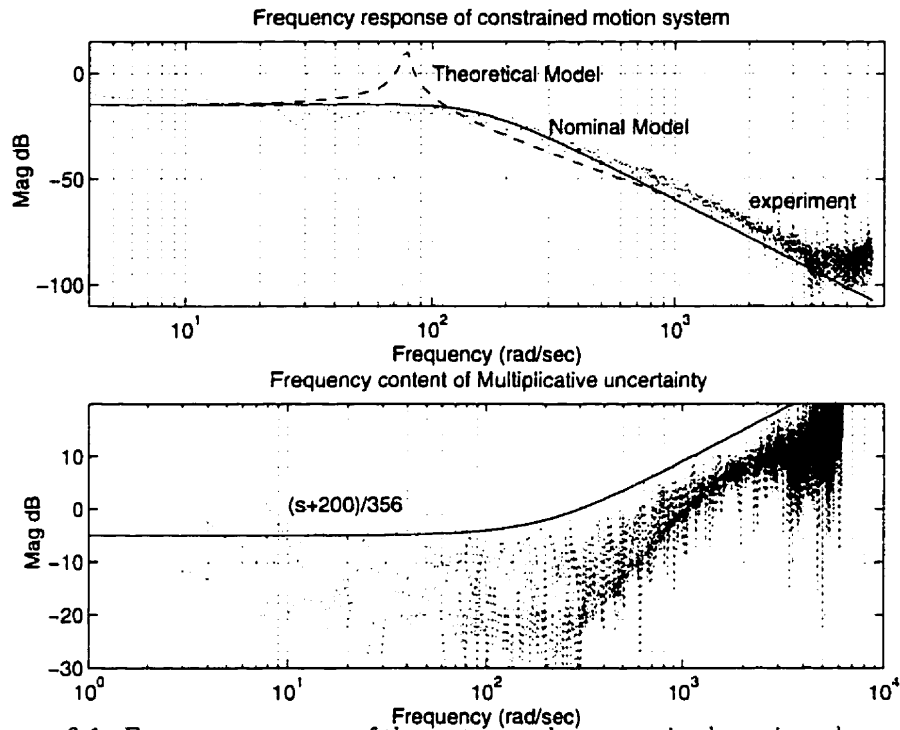


FIGURE 6.1. Frequency response of the system under constrained-motion, theoretical and nominal models, and multiplicative uncertainty

be a third order stable and minimum phase transfer function as follows:

$$\frac{\text{Torque}}{\text{Ref Voltage}} = \frac{1.0755 \times 10^6}{s^3 + 472.7s^2 + 7.33 \times 10^4s + 5.89 \times 10^6} \quad (6.4)$$

This system has three stable poles at -289.83 and $-91.44 \pm 109.48j$ and a DC-gain of -14.8 dB. Using Equation 6.3 the system variations for four typical frequency response estimates is illustrated in Figure 6.1, and the uncertainty weighting function is approximated by $W(s) = (s + 200)/356$.

1.2. System under Free-Motion

Similar to the constrained-motion case, an empirical nominal model for the system is derived using experimental frequency response data of the system for free-motion experiments. Figure 6.2 illustrates the empirical frequency responses of the system under free-motion, including nominal model and its uncertainty bound. The nominal model for the system is found to be a third order stable and minimum phase transfer function as follows:

$$\frac{\text{Torque}}{\text{Ref Voltage}} = \frac{243.16 (s + 2.415)}{s^3 + 171.19s^2 + 1.24 \times 10^4s + 1.47 \times 10^5} \quad (6.5)$$

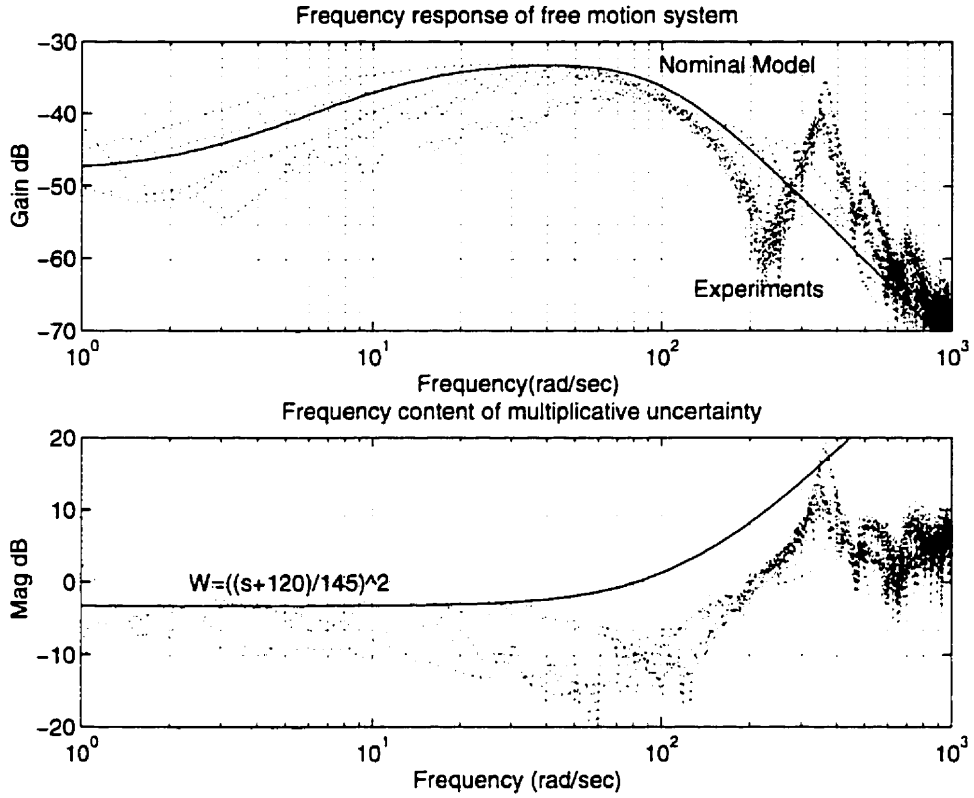


FIGURE 6.2. Frequency response of the free-motion system, nominal model, and multiplicative uncertainty

which has three stable poles at -14.465 , and $-78.363 \pm 63.288j$, and a DC-gain of -48 dB. The uncertainty weighting function is approximated by a second order system as $\mathbf{W}(s) = \left(\frac{s+120}{145}\right)^2$. Note that the lower DC-gain in free-motion system is due to smaller torque outputs in free-motion experiments when compared to the constrained-motion case. Also the system variations in free-motion are larger than that in the constrained-motion, since the nonlinear friction plays a more important role for low-frequency free-motion experiments. These two factors make the control of free-motion case harder than that in the constrained-motion case. Moreover, the resonance/anti-resonance feature of the empirical frequency response of the free-motion system, observed at frequencies about 200 and 400 (rad/sec) of Figure 6.2, represents the typical higher mode vibration of the flexspline. The nominal model of the system is not representing the higher mode vibration of the system, and as elaborated in § 2.3.1 of Appendix A, these unmodelled dynamics are encapsulated by an increasing uncertainty at high frequencies.

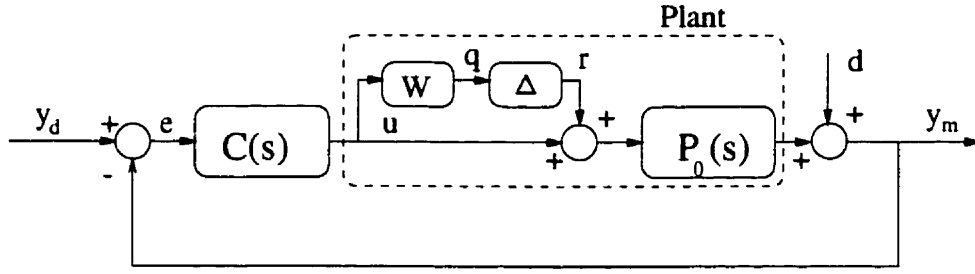


FIGURE 6.3. Block diagram of the system with multiplicative uncertainty for plant

2. Control Objectives

Figure 6.3 illustrates the block diagram associated with the multiplicative uncertainty representation of Equation 6.1. P_0 is the nominal model of the system, W is the uncertainty weighting function, Δ is a variable $\mathcal{H}_\infty$ transfer function of norm less than unity, and C is the controller. The control objective can be defined as *robustly stabilize the system, while maintaining good disturbance attenuation and small tracking error, despite actuator saturation*. More specifically, referring to Figure 6.3, we would like to design a controller to trade-off minimizing the norm of the transfer function from reference input y_d to the tracking error e (tracking performance), the transfer function from the disturbance d to the output y (disturbance attenuation), the transfer function from r to q (robust stability), and the transfer function from reference input y_d to the plant input u (actuator limit). This multi-objective problem is well-suited to the general $\mathcal{H}_\infty$ -framework.

3. Controller Design

As explained in § 2.6 of Appendix A the multi-objective problem defined in the preceding section can be reformulate into a mixed-sensitivity problem. Figure 6.4 illustrates the block diagram of the system configured as a single-input multi-output (SIMO) for the $\mathcal{H}_\infty$ -framework. As illustrated in Appendix A, tracking and disturbance attenuation objectives can be expressed as sensitivity $\mathbf{S}$ minimization [12]. For multiplicative uncertainty robust stability is guaranteed if the complementary sensitivity $\mathbf{T}$ has a norm less than unity (Small Gain Theorem [126]). $\mathbf{T}$ can be shown to be the transfer function from reference input y_d to the output y . Weighting functions W_s and W_u are also considered to normalize and assign frequency content to the performance objectives on sensitivity and motor current saturation respectively, and W is the multiplicative uncertainty weighting function. The

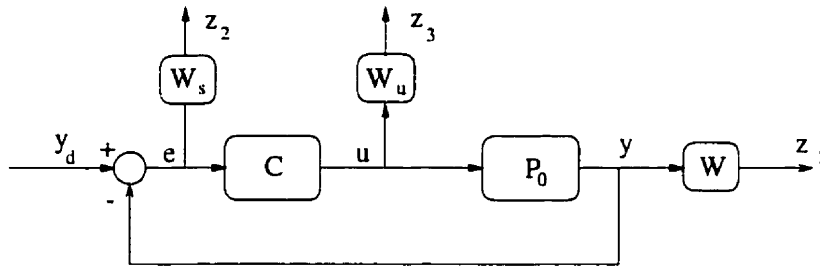


FIGURE 6.4. Block diagram of the reconfigured system of Figure 6.3 suited for an $\mathcal{H}_\infty$ -framework

augmented system has one input y_d , and three outputs z_1 , z_2 , and z_3 , in which the transfer function from the input to the outputs corresponds to weighted complementary sensitivity, weighted sensitivity, and weighted actuator effort, respectively.

The objectives now will be reduced to finding a controller $C(s)$ which minimizes the induced norm of the transfer matrix from input y_d to the output vector $\mathbf{z}$ or,

$$\text{Find } C(s) \text{ which minimizes } \|\mathbf{T}_{y_d \mathbf{z}}\|_\infty. \quad (6.6)$$

This problem is called a mixed-sensitivity problem in the literature, and has optimal and sub-optimal solution algorithms. Doyle et al. [35], provided the sub-optimal solution for this problem in 1989, in which $C(s)$ will be selected such that $\|\mathbf{T}_{y_d \mathbf{z}}\|_\infty < 1$. The μ -synthesis toolbox of Matlab uses this algorithm iteratively to find the best sub-optimal solution achievable [10].

Performance-weighting functions are selected considering the physical limitations of the system. The actuator saturation-weighting function $\mathbf{W}_u$ is considered to be a constant, so that the maximum expected input never saturates the actuator. Its value is estimated to be 0.004 for the system under constrained-motion, and 0.002 for free-motion.

The sensitivity weighting function for the constrained-motion setup is assigned to be $\mathbf{W}_s(s) = \frac{s+300}{2(s+3)}$. This weighting function indicates that at low frequencies, the closed-loop system should reject disturbance at the output by a factor of 50 to 1. Expressed differently, steady-state tracking errors due to step input should be less than 2 %. This performance requirement becomes less stringent with higher frequencies. For high frequencies the closed-loop frequency response should degrade gracefully, always lying underneath the inverse of the weighting function $\mathbf{W}_s$.

For free-motion setup the sensitivity weighting function is assigned to be $\mathbf{W}_s(s) = \frac{s+280}{5(s+2.8)}$, which permits 5 % steady state tracking error for the closed loop system. The separate choice of sensitivity weighting function for free-motion and constrained-motion

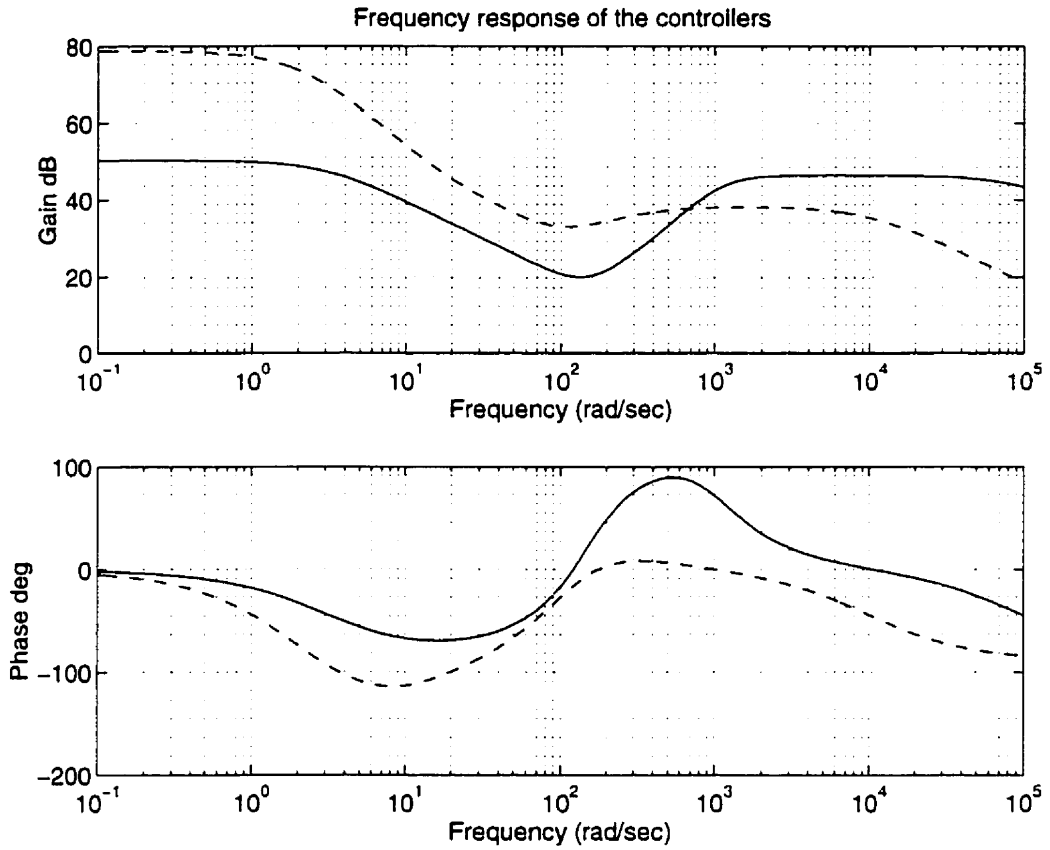


FIGURE 6.5. The frequency response of the two designed controllers; Solid : Constrained-motion, Dashed : Free-motion

permits us to have similar bandwidth characteristics for the closed loop systems despite the lower torque output and signal-to-noise ratio observed in the free-motion case. For both cases the best cut-off frequency for performance is found by an iterative method, provided the $\mathcal{H}_\infty$ solution to the problem exist.

Two controllers were designed using the μ -synthesis toolbox of Matlab. For constrained-motion case the resulting controller transfer function is:

$$C_{CM}(s) = \frac{2.08 \times 10^7 (s + 289.8)(s + 91.4 \pm 109.5j)}{(s + 3)(s + 808.2 \pm 776.04j)(s + 9.8 \times 10^4)} \quad (6.7)$$

with a DC-gain of 50.4 dB, while for free-motion setup the resulting controller transfer function is as follows with a DC-gain of 78.8 dB.

$$C_{FM}(s) = \frac{8.345 \times 10^5 (s + 14.5)(s + 78.4 \pm 63.3j)}{(s + 1.83)(s + 2.8)(s + 273.23)(s + 1.0 \times 10^4)} \quad (6.8)$$

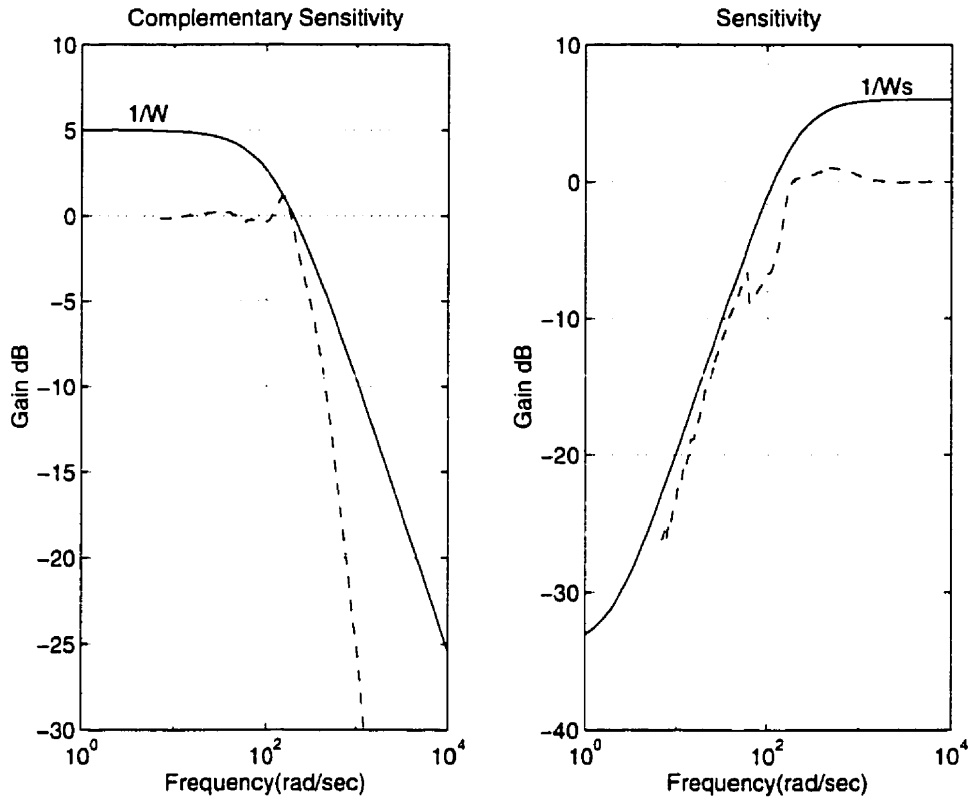


FIGURE 6.6. Closed-loop frequency performance of the system under constrained-motion with weighting envelopes

The controller zeros cancel the stable poles of the nominal plant, while the poles shape the closed-loop sensitivity function. Figure 6.5 illustrates the Bode plot of the two controllers, where for both controllers there is a wide anti-resonance profile, to shape the complementary sensitivity function as flat as possible. Hence, it is not possible to obtain similar performance through a PID, or lead-lag controller. Furthermore, the free-motion controller has larger DC-gain to compensate for the comparatively lower measured torque signal.

4. Closed-loop Performance

To verify the controller performance, closed-loop experiments are employed. To implement the controllers in practice, bilinear discretization on Equations 6.7 and 6.8 is performed with one kHz sampling frequency. The performance of the closed-loop system under constrained-motion and free-motion is evaluated in both frequency and time domain for

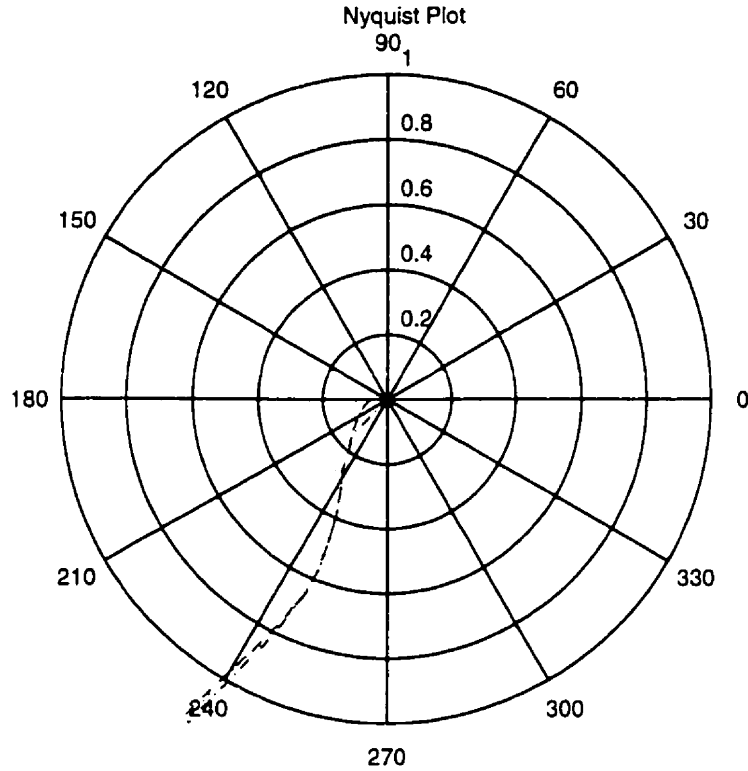


FIGURE 6.7. Nyquist plot of the system under constrained-motion obtained from empirical frequency response of the closed loop system

the McGill and ISE setups. However, because of the similarity in results, here only the experimental results of the McGill setup is presented. The results of ISE setup can be found in [107].

4.1. System under Constrained-Motion

4.1.1. Frequency Domain Performance

The frequency domain performance of the closed-loop system is obtained from the closed-loop frequency response experiments and is illustrated in Figure 6.6. For both setups the experimental sensitivity and complementary sensitivity functions are shown to be underneath the inverse of $\mathbf{W}_s$, and $\mathbf{W}$ respectively. Also the Nyquist plot for the loop-gain of the system is derived from the experimental closed-loop frequency response, and illustrated in Figure 6.7. The gain margin is found to be 18.2, while phase margin is 60° . Hence, very good stability margins are obtained with the robust torque controller. These results provide an experimental verification of the $\mathcal{H}_\infty$ design claim to preserve robust stability while shaping the performance.

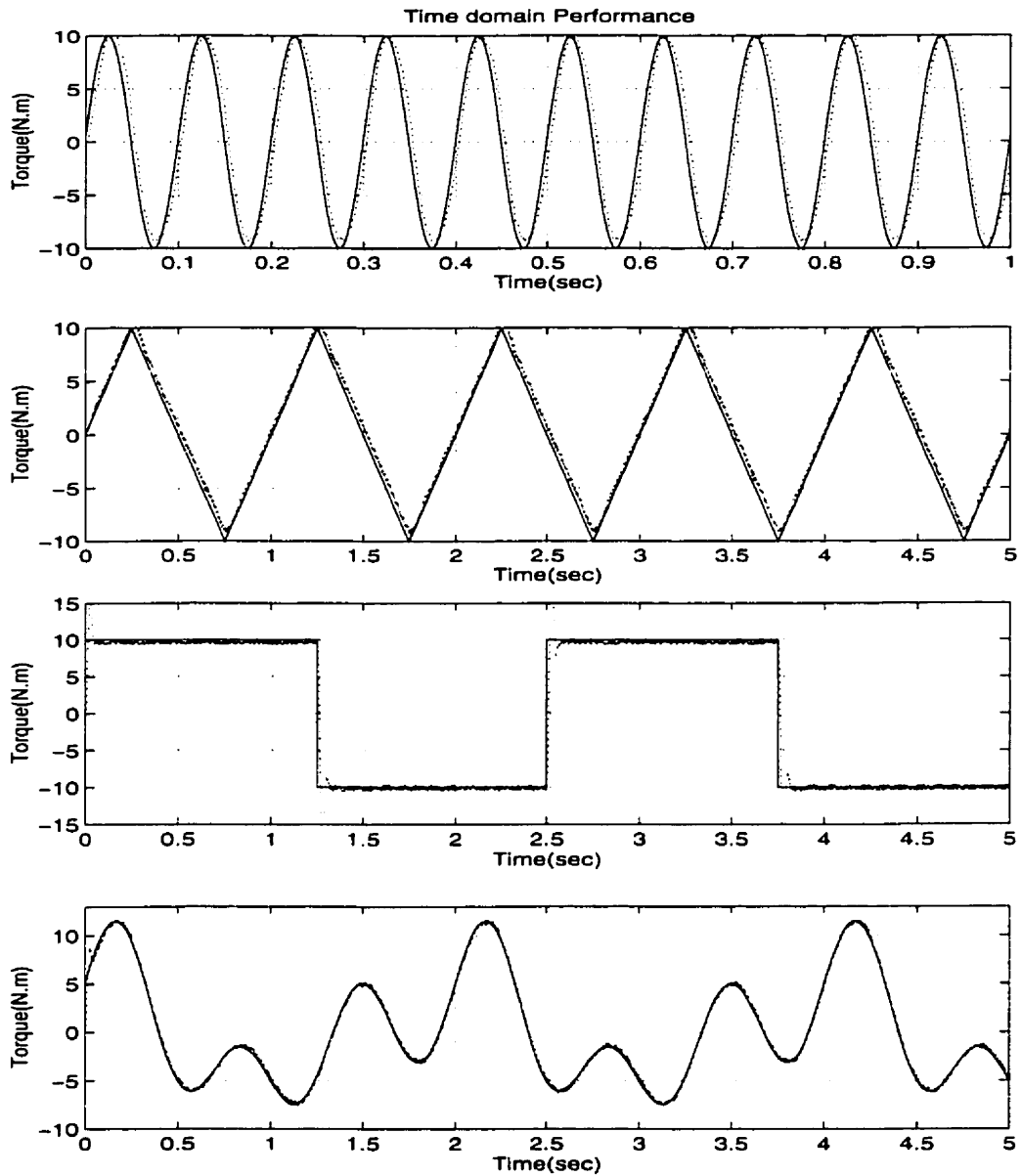


FIGURE 6.8. Closed-loop time response of the system under constrained-motion

4.1.2. Time Domain performance

The time response of the closed-loop system to different reference input signals is illustrated in Figure 6.8. The dotted lines are the measured output torque of the system, which are tracking the solid reference command, very fast and accurately. Although our designed bandwidth is 3 rad/sec, sinusoid inputs up to 10 Hz (62 rad/sec) are shown to be tracked. The step response is very fast with a 15 ms rising time and a steady-state error

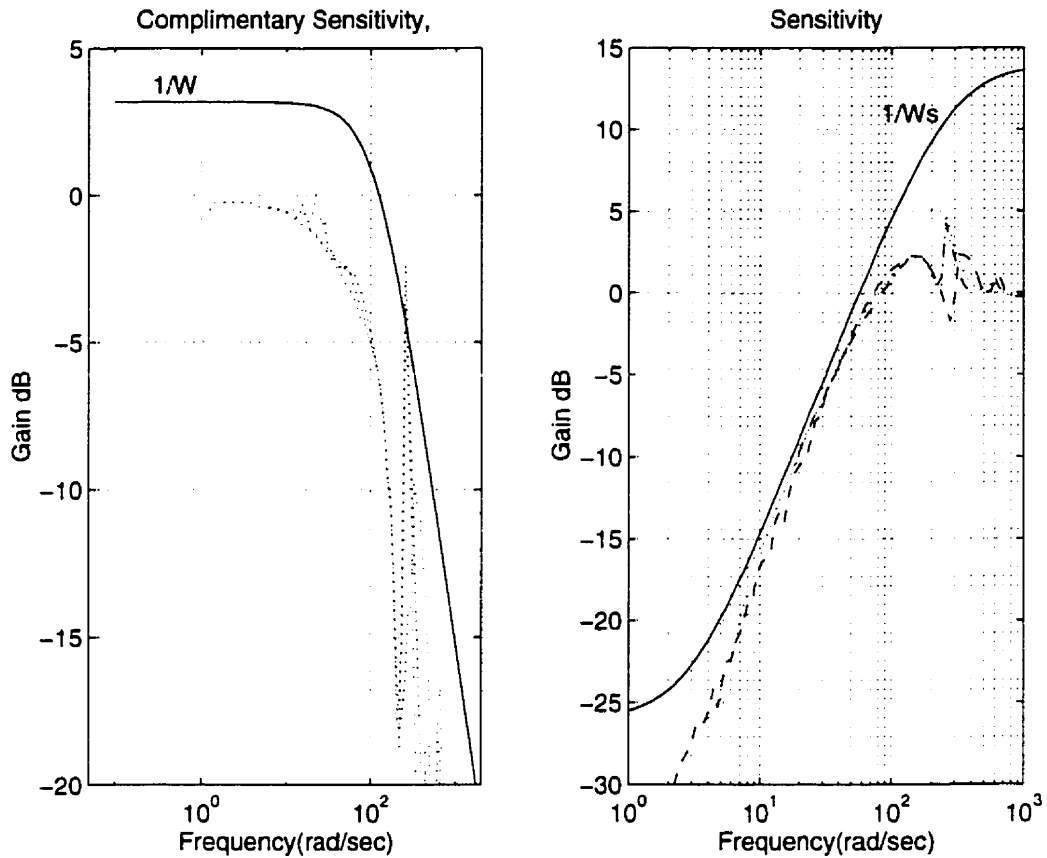


FIGURE 6.9. Closed-loop frequency performance of system under free-motion

less than 2% as required. Tracking of the system to triangular signal is especially sharp at the edges, and the tracking to an generic signal is shown to be very fast and well-behaved.

4.2. System under Free-Motion

4.2.1. Frequency Domain Performance

The frequency domain performance of the closed-loop system is obtained from the closed-loop frequency response of the system and is illustrated in Figure 6.9. The experimental sensitivity and complementary sensitivity functions are shown to be underneath the inverse of W_s , and W respectively. Also the Nyquist plot for the loop-gain of the system is derived from the experimental sensitivity functions. Figure 6.10 illustrates the Nyquist plot of the system, in which the gain margin is 6.2, and phase margin is 80° . Hence, very good stability margins are obtained by means of robust torque controller.

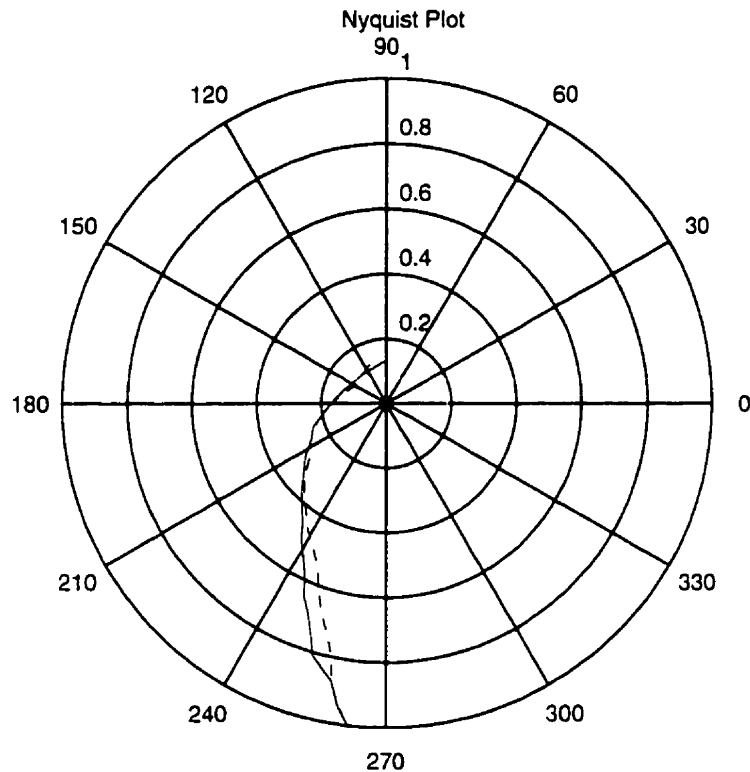


FIGURE 6.10. Nyquist plot of the system under free-motion obtained from empirical frequency response of the closed loop system

4.2.2. Time Domain performance

The time response of the closed-loop system to different reference input signals is illustrated in Figure 6.11. The dotted lines are the measured output torque of the system, which is tracking the solid line, the reference command. Although our designed bandwidth is about 2.8 rad/sec, sinusoid inputs up to 10 Hz (62 rad/sec) are shown to be tracked relatively well. The step response is very fast with a rising time of 20 ms and tracking of the system to triangular signal is especially sharp at the edges. Finally, tracking to an arbitrary signal is shown to be very fast and well-behaved.

The performance of the closed-loop system under free-motion case are not as good as that in constrained-motion, because in constrained-motion experiments the open-loop system has higher DC-gain, lower uncertainty at low frequencies, and higher signal-to-noise ratio. Hence, wider bandwidth and better closed-loop performance are achievable as illustrated in Figure 6.8.

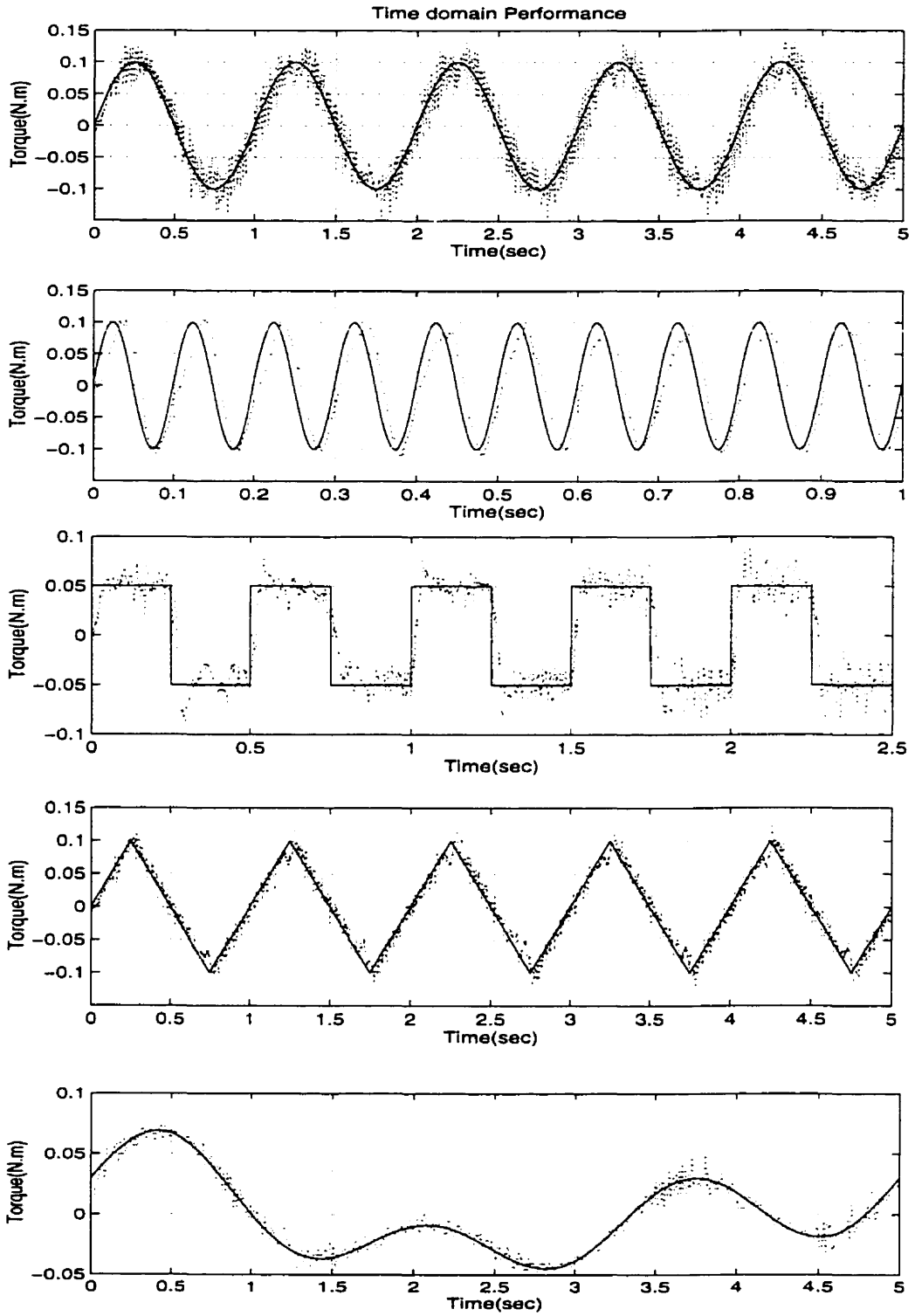


FIGURE 6.11. Closed-loop time response of the system under free-motion

5. Friction-Compensation

To improve the closed-loop performance of the system under free-motion, we applied a friction-compensation algorithm to the open-loop system. As illustrated in Figure 6.2, the frequency response of the system under free-motion possess significant variations at low frequencies. This is mainly caused by the nonlinear behaviour of friction which is dominant at low frequencies. In § 3.2 of Chapter 4, a complete model of harmonic drive friction was presented as Coulomb, viscous and Stribeck friction [103]. Friction parameters were carefully identified from experiments. As illustrated in § 3 of Chapter 5, the effect of Coulomb and Viscous friction is significant for free-motion experiments, while Stribeck friction remains apparent in low-velocity experiments. Therefore, in the sequel study we compensate for Coulomb and viscous friction only.

5.1. Friction-Compensation Algorithm

From Equation 4.17, the equation governing the harmonic drive friction can be modified to:

$$T_{fric}(\dot{\theta}) = T_{v_n} u_{-1}(-\dot{\theta})\dot{\theta} + T_{v_p} u_{-1}(\dot{\theta})\dot{\theta} + T_{s_n} u_{-1}(-\dot{\theta})\text{sign}(\dot{\theta}) + T_{s_p} u_{-1}(\dot{\theta})\text{sign}(\dot{\theta}) \quad (6.9)$$

where

$$u_{-1}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \quad (6.10)$$

T_{v_n} and T_{v_p} are the viscous friction coefficient depending on the direction of the velocity, and T_{s_n} and T_{s_p} are the Coulomb friction coefficients as illustrated in Figure 6.12. For the McGill setup the identified parameters are: $T_{v_n} = 3.5 \times 10^{-4}$, $T_{v_p} = 3.7 \times 10^{-4}$ ($N.m.sec/rad$), $T_{s_n} = 4.4 \times 10^{-2}$, and $T_{s_p} = 4.6 \times 10^{-2}$ ($N.m$) as given in Table 4.2.

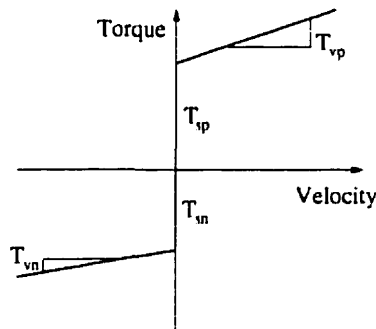


FIGURE 6.12. Identified Coulomb and viscous friction curve for harmonic drive systems

The idea of friction-compensation is to estimate the friction torque at each instant from the measured velocity of the system, and to increase the reference command to the servo-amp corresponding to the estimated friction. Ideally, estimated friction should be equal to the actual friction; however, the magnitude of the friction depends on the operating condition, and special care must be taken so that over-compensation does not occur, which introduces instability into the system. For our experimental setup only 90 % of the estimated friction is compensated in order to avoid over-compensation, as suggested by Kubo et al. [78]. Another practical issue in the friction-compensation algorithm is the method of implementing hard nonlinear Coulomb friction. The estimated friction will change sign as velocity crosses zero. In practice, however, the velocity measurement is sampled, and hence, zero velocity crossing may never coincide at the sampling instants. Moreover, the velocity signal is always contaminated with noise, and at low velocity several unrealistic zero crossing may appear. To avoid chattering in friction-compensation, a *threshold velocity* was introduced in the literature [1, 9, 70] to smooth the hard nonlinearity of Coulomb friction. Including the threshold velocity $\dot{\theta}_t$ the final friction estimation function will be as follows:

$$T_{fric}(\dot{\theta}, \dot{\theta}_t, V_{ref}) = \begin{cases} T_{vp}\dot{\theta} + T_{sp} & \dot{\theta} > \dot{\theta}_t \\ T_{vp}\dot{\theta} + T_{sp} & |\dot{\theta}| \leq \dot{\theta}_t \text{ \& \& } V_{ref} > 0 \\ T_{vn}\dot{\theta} - T_{sn} & |\dot{\theta}| \leq \dot{\theta}_t \text{ \& \& } V_{ref} < 0 \\ T_{vn}\dot{\theta} - T_{sn} & \dot{\theta} < -\dot{\theta}_t \end{cases} \quad (6.11)$$

in which V_{ref} is the reference voltage commanded to the servo-amp, and the threshold velocity is set to $\dot{\theta}_t = 1 \text{ (rad/sec)}$ for the experiments. Within the threshold velocity region the direction of the friction torque is determined by the sign of the reference command signal instead of the velocity. It is verified by experiments that this representation of the friction torque at low velocities eliminates the chattering problem.

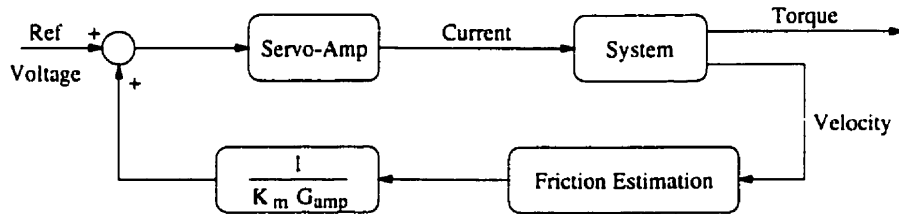


FIGURE 6.13. Block diagram of the friction-compensation algorithm

Figure 6.13 illustrates the Block diagram of friction-compensation algorithm implemented on the setup. The corresponding reference command compensating for the friction is estimated by dividing the friction torque estimate by the motor torque constant $K_m(N.m/amp)$ and by the servo-amp gain $G_{amp}(amp/volt)$, as illustrated in Figure 6.13.

5.2. System Model and its Reduced Uncertainty

Similar to the free-motion case, an empirical nominal model for the system including the friction-compensation can be derived using experimental frequency response. Figure 6.14 illustrates the empirical frequency responses of the system with friction-compensation, including nominal model and its uncertainty weighting function. The effect of friction-compensation on the variation of the frequency response estimates at low frequencies is clearly seen when compared to Figure 6.2. The friction-compensated system behaves more linearly at low frequencies, and hence, the uncertainty of the system shrinks at low frequencies, from -3.3 dB to -10 dB. The uncertainty measure of the system is therefore, not only

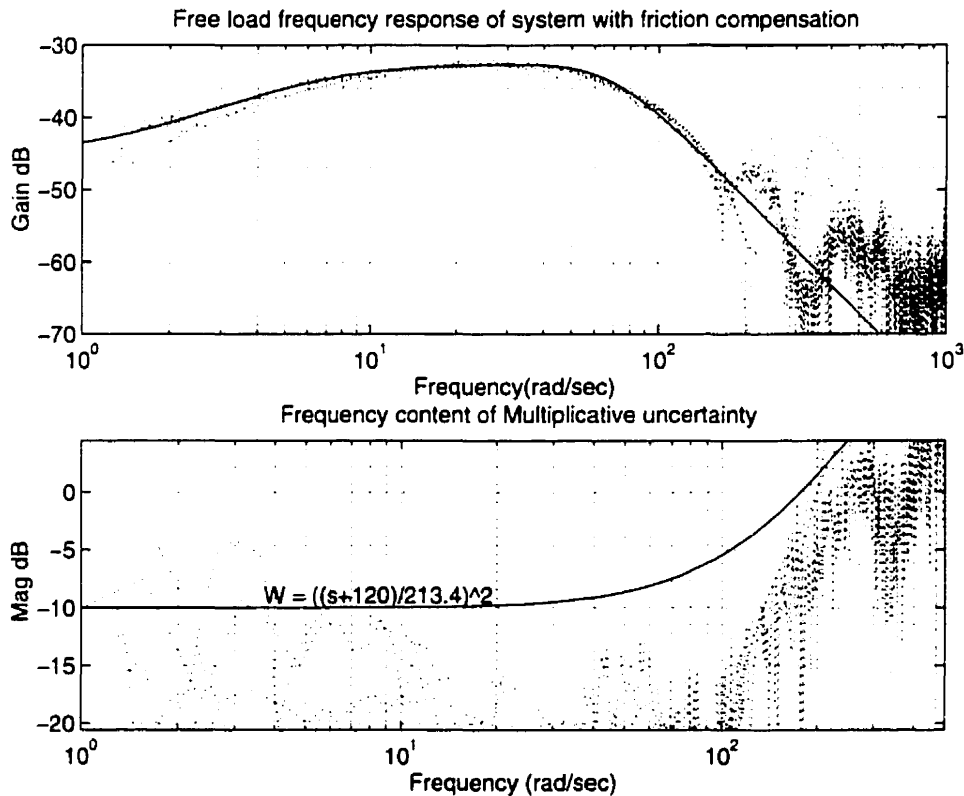


FIGURE 6.14. Frequency response of the free-motion system with friction compensation, its nominal model, and multiplicative uncertainty

used in $\mathcal{H}_\infty$ synthesis to design the controller, but also as a gauge of the effectiveness of the friction-compensation algorithm.

A nominal model for the friction-compensated system is taken to be the following third-order stable and minimum-phase transfer function

$$\frac{\text{Torque}}{\text{Ref Voltage}} = \frac{109.4 (s + 1.363)}{s^3 + 96.06s^2 + 5159s + 2.71 \times 10^4} \quad (6.12)$$

which has three stable poles at -5.868 , and $-45.096 \pm 50.953j$, and a DC-gain of -45.2 dB. The uncertainty weighting function is approximated by $\mathbf{W}(s) = \left(\frac{s+120}{213.4}\right)^2$.

5.3. Robust Torque Control

Similar to the free-motion case, for the friction-compensated system a controller is designed using an $\mathcal{H}_\infty$ -framework. The sensitivity weighting function is assigned $\mathbf{W}_s(s) = \frac{s+530}{5(s+5.3)}$ with a bandwidth of 5.3 (rad/sec), which compared to 2.8 (rad/sec) bandwidth in free-motion system is a significant improvement. The 5 % steady state tracking error is maintained, while in another trial, it has been shown that 2 % tracking error assignment is also achievable, but with a bandwidth of 1.75 (rad/sec). The actuator saturation-weighting function is set to be 0.002, the same as was assigned in system without friction-compensation. The controllers were designed using the μ -synthesis toolbox of Matlab by solving the mixed-sensitivity problem explained in § 5.3. For friction-compensated system controller transfer function is:

$$\mathbf{C}_{FC}(s) = \frac{3.30 \times 10^6 (s + 5.8679)(s + 45.10 \pm 50.95j)}{(s + 1.27)(s + 5)(s + 318.86)(s + 1.06 \times 10^4)} \quad (6.13)$$

with a DC-gain of 72 dB.

5.4. Closed-Loop Performance Comparison

To compare the performance of the closed-loop system with and without friction-compensation, frequency domain sensitivity and complementary sensitivity functions are shown in Figure 6.15. The friction-compensated system has a smaller sensitivity function at low frequencies, compared to the system without friction-compensation; however, its complementary sensitivity function shows larger overshoot close to the resonance frequency. Nevertheless, this is well below the inverse of the uncertainty weighting function, and hence, the robust stability has not deteriorated. The performance comparison of the system in time-domain is illustrated in Figure 6.16. For low-frequency sinusoid and triangular signals friction-compensation has improved the performance while for signals with high frequency content the performance is not improved, as illustrated for 10 Hz sinusoid and squared signals. Over all, for the applications where signals with low frequency content

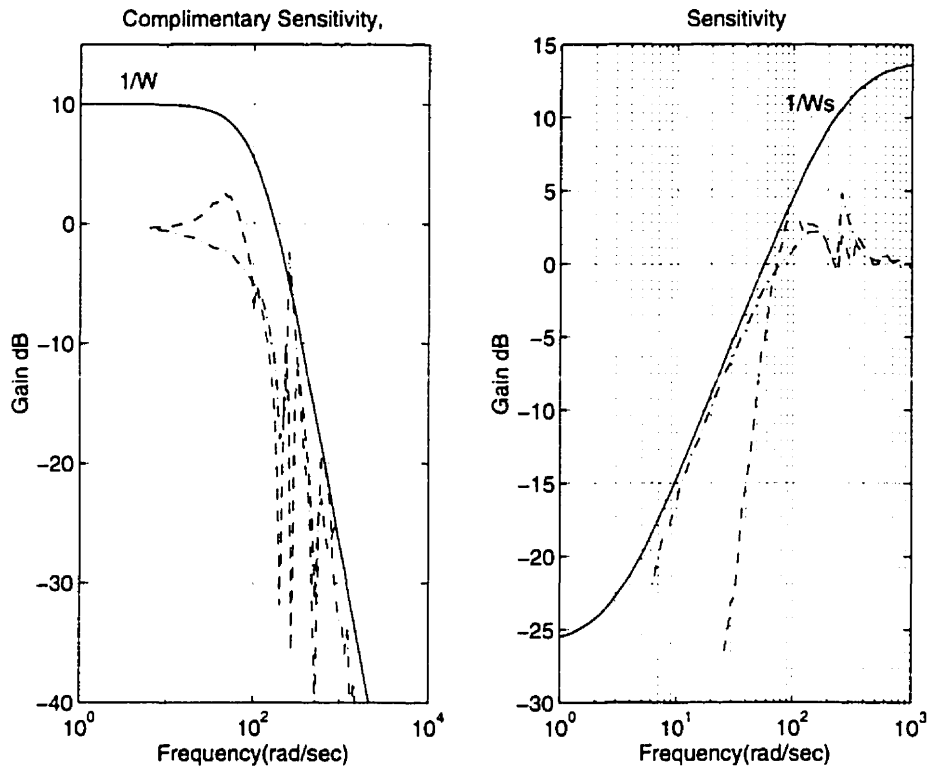


FIGURE 6.15. Closed-loop frequency performance comparison of the system with and without friction-compensation; Dashed : With friction-compensation, Dash-Dotted : Without friction-compensation

should be tracked, this comparison suggest that the friction-compensation will result in a superior performance. However, for the applications where high frequencies are involved (step response for instance) friction-compensation doesn't contribute to performance. This is because as illustrated in Figure 6.14 friction-compensation will linearize the system only at low-frequencies, and moreover, a percentage of the servo-amp power is consumed for the friction-compensation, so less power is available for high frequency trackings. In our experimental setup, depending on the output velocity, 12 – 25% of the servo-amp power was utilized for friction-compensation algorithm.

6. Torque Ripple Cancellation

One important characteristic of harmonic drive torque transmission in free motion experiments, is a high frequency oscillation in the output torque called *torque ripples*. In § 3 of Chapter 3 we introduced the torque ripples and presented a methodology to estimate and filter them from the torque measurement, and the filtered torque is employed as an

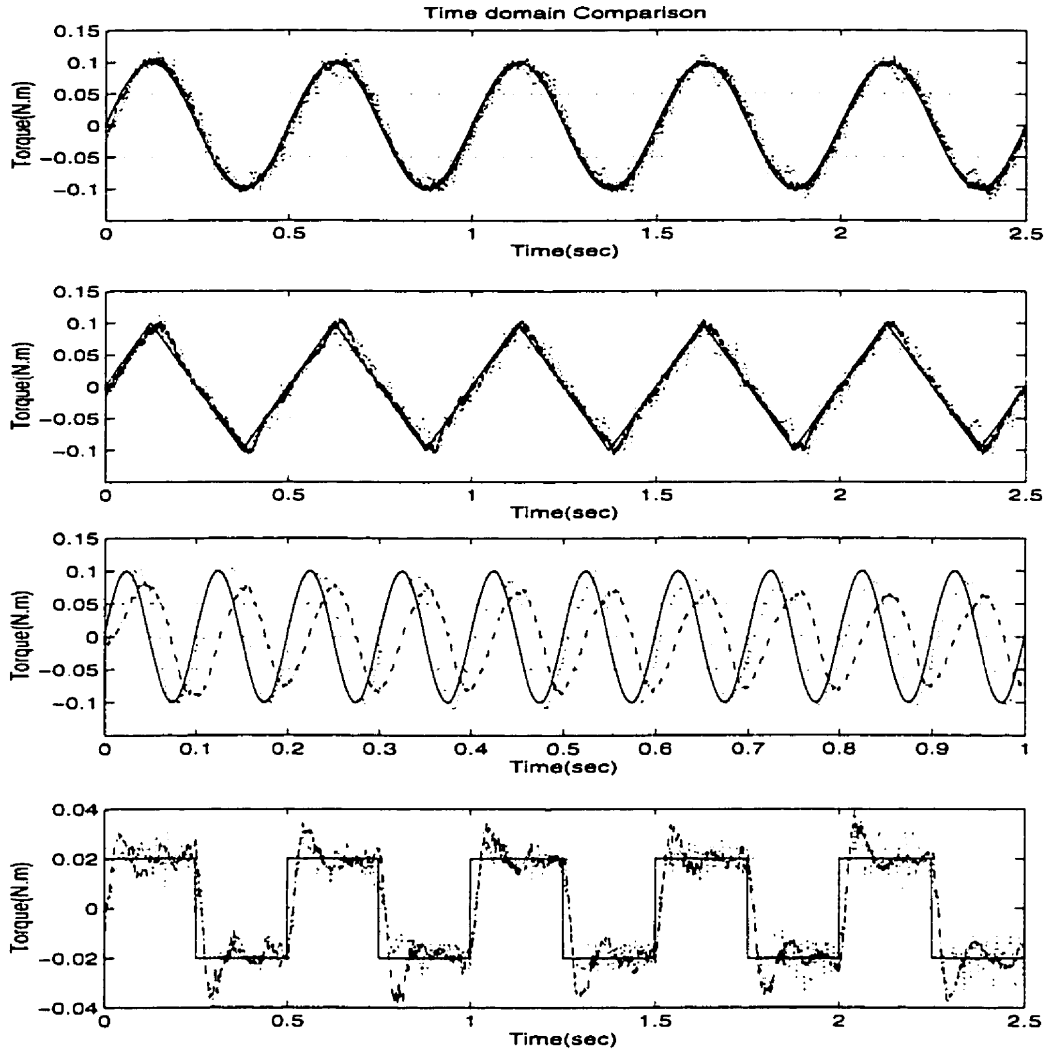


FIGURE 6.16. Closed-loop time response comparison of the system with and without friction-compensation; Solid : Reference command, Dash-Dotted : With friction-compensation, Dotted : Without friction-compensation

integral part of the torque-feedbacks designed in § 3. However, torque ripples are real torque oscillations, caused mainly by harmonic drive gear meshing vibration. They can be observed in the end effector of robots using harmonic drives and even sensed by hand when back-driving the harmonic drive. Hence, in some applications where the smoothness of the output torque are important, it is desirable to attenuate the torque ripples.

The effect of the torque ripples on the output inertia can be reduced by a control algorithm, or by designing a vibration isolation system or a vibration absorber. Let us first examine the possible control algorithm which can cancel or reduce the torque ripples.

This could be realized through an algorithm similar to the friction compensation routine discussed in § 5, in which the value of the torque ripple at each instance is estimated and the reference command to the servo-amp is adapted with respect to the torque ripple estimate. It is shown in Chapter 3 that torque ripples can be accurately estimated using Kalman filter estimation [110]. To implement the control algorithm, however, some practical issues should be addressed. As illustrated in Figure 3.6 the amplitude of the torque ripples varies from 1 to 1.5 $N.m$, while for nominal operating velocity of the system the frequency content of the torque ripples has a peak at a frequency of about 600 rad/sec . Hence, the power required to compensate this high frequency oscillations varies between 600 to 900 $watts$. This requirement exceeds the 100 $watts$ of power that our power amplifier can deliver. This shortcoming can be overcome through a careful design process in which the amplitude of the oscillation is reduced to a minimum using a vibration isolator, or a vibration absorber, while still providing a high-power wide-bandwidth amplifier and matching DC-motor to be able to cancel the torque ripples completely. This process is not implemented in our setup because of its cost. Nevertheless, the technical information and general guidelines for designing an vibration isolator, and vibration absorber are given in Sections 6.1 and 6.2 as a future reference.

6.1. Vibration Isolation

The most effective way to reduce unwanted oscillations is to stop or modify the source of the vibration. For harmonic drive systems, however, the source of vibration is the flexibility of the flexspline itself which amplifies the gear meshing vibration. Since this is the inherent characteristics of the harmonic drive system, it is not possible to eliminate the source of the vibration. But it is possible to design a *vibration isolation system*, which is indeed a mechanical low-pass filter, to isolate the flexspline vibration from the output load. This can be done by using damping materials such as rubber between the flexspline and output load. By this means the stiffness and damping characteristics between the flexspline and the load are changed. Hence, this is recommended strictly for applications where the overall stiffness characteristics of the harmonic drive system is not crucial. Vibration isolation is analyzed in terms of reducing the vibration transmitted from the flexspline through the output load. This can be quantified by the *transmissibility ratio*, $T.R.$, defined as *the ratio of the magnitude of the torque transmitted to the load to the sinusoidal torque generated by the vibration source in the flexspline at steady state*. Figure 6.17 illustrates a vibration isolator modelled as a linear spring k and a viscous damper c connected in parallel between the flexspline and the load each modelled as a simple inertia. T_{fs} is the torque ripple

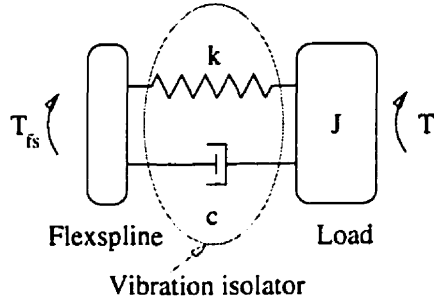


FIGURE 6.17. The model of vibration isolator as a spring and damper

generated by the flexspline gear meshing vibration, while T_t is the torque transmitted to the load. Consider the main or the first-harmonic of the torque ripple as: $T_{fs} = T_o \sin(\omega t)$, with frequency ω , and torque amplitude of T_o . The transmissibility ratio can be calculated from, [66]:

$$T.R. = \frac{T_t}{T_o} = \left[\frac{1 + (2\zeta r)^2}{(1 - r^2)^2 + (2\zeta r)^2} \right]^{1/2} \quad (6.14)$$

in which the damping ratio ζ , the frequency ratio r , and the natural frequency ω_n are defined from the characteristics of the load inertia J , vibration isolator stiffness and damping coefficients via

$$\omega_n = \sqrt{\frac{k}{J}} \quad , \quad r = \frac{\omega}{\omega_n} \quad , \quad \zeta = \frac{c}{2J\omega_n} \quad (6.15)$$

Figure 6.18 illustrates the transmissibility ratio in logarithmic scale versus the frequency ratio for a variety of damping ratios ζ . These curves are useful in designing isolators. In particular, the design process consists in choosing ζ and r , within the available isolator's material, such that $T.R.$ is as small as possible. In Figure 6.18, values of the transmissibility ratio divide parameter space into two regions. For $r < \sqrt{2}$, $T.R. > 1$, where in this region the load vibration is magnified with respect to that of the flexspline. For $r > \sqrt{2}$ however, $T.R. < 1$ and this region is called *isolation region*. In this region the smaller the value of ζ , the smaller the value of $T.R.$. However, some damping is desirable in order to reduce the transmissibility at resonance ($r=1$).

Although Equation 6.14 is derived using the simplifying assumption of linearity in stiffness and damping in the isolators, it provides insight through design or selection of the vibration isolators in practice. In the process of design, the natural frequency of the isolator used in the system ω_n , should be selected such that, within the operating velocity of the system, r remains always greater than $\sqrt{2}$. Depending on design limitation selecting

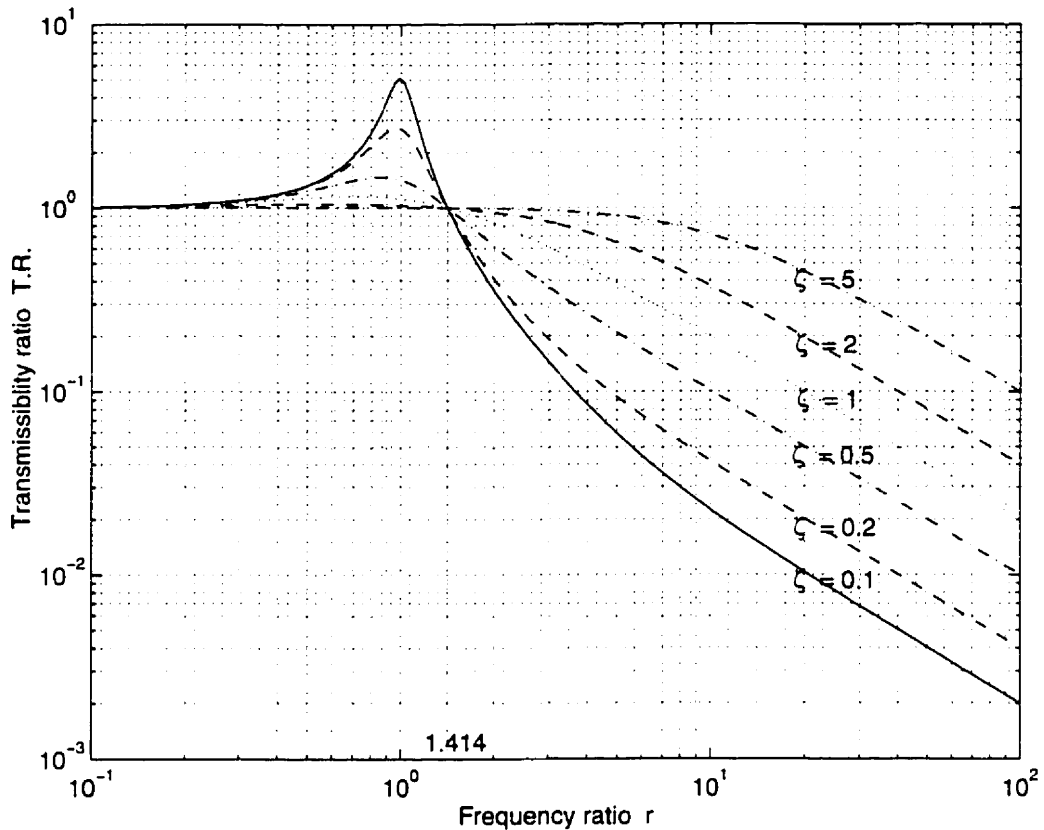


FIGURE 6.18. Plot of the transmissibility ratio in logarithmic scale for a vibration isolator including a variety of damping ratios ζ

a large value of r (i.g. $r = 10$) with some reasonable choice of ζ (i.g. $\zeta = 0.2$) can reduce the amplitude of the torque ripples significantly. Commercial isolators are widely manufactured and used in industrial applications [26, 80]. Most commercial isolators are made of elastomers, a term used to describe viscoelastic material made of rubber, either synthetic and natural.

6.2. Vibration Absorbers

Another approach to reduce the load oscillation due to the torque ripples, is by means of *vibration absorber* which is indeed a mechanical notch filter. Unlike the isolator of the previous section, an absorber consists of a second inertia-spring system coupled to the load inertia to protect it from vibrating. The absorber inertia and stiffness are designed such that the motion of the load inertia becomes a minimum. This is accompanied by substantial motion of the added absorber system. Figure 6.19 illustrates a simple vibration absorber modelled as an inertial-spring attached to a load inertia. By modelling the stiffness of the

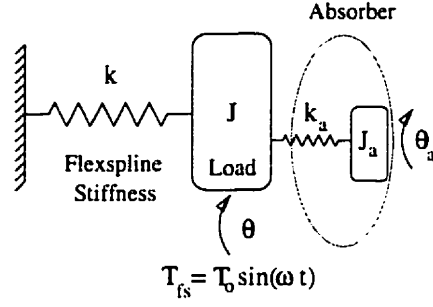


FIGURE 6.19. The model of a vibration absorber as an inertia and a spring

flexspline as a linear spring and considering the torque acting on the inertia to be only the main or the first harmonic of the torque ripple, the whole system is viewed as a two-degree-of-freedom system excited by a sinusoid input $T_{fs} = T_o \sin(\omega t)$. It is desired to assign the absorber parameters (i.e. the inertia J_a , and the stiffness k_a) such that the oscillation of the load inertia in steady state is as small as possible. The steady state solution of the oscillation of the load θ , and the absorber θ_a can be shown to be sinusoids with amplitude θ_o and θ_{ao} given by [66]:

$$\theta_o = \frac{(k_a - J_a \omega^2) T_o}{(k + k_a - J \omega^2)(k_a - J_a \omega^2) - k_a^2} \quad (6.16)$$

$$\theta_{ao} = \frac{k_a T_o}{(k + k_a - J \omega^2)(k_a - J_a \omega^2) - k_a^2} \quad (6.17)$$

Equation 6.16 shows that the absorber parameters k_a , and J_a can be chosen such that the amplitude of steady state vibration, θ_o becomes zero. This is accomplished by equating the coefficient of T_o in Equation 6.16 to zero:

$$k_a = J_a \omega^2 \quad (6.18)$$

In this case the steady state motion of the absorber is:

$$\theta_a(t) = -\frac{T_o}{k_a} \sin(\omega t) \quad (6.19)$$

Note that the amplitude of the torque acting on the absorber is $k_a \theta_{ao} = -T_o$. Hence when the absorber system is tuned to the driving frequency and has reached steady state, the torque provided by the absorber system is equal in magnitude and opposite in direction to the torque ripples. With zero net torque acting on the load, the motion of the load is “absorbed” by the motion of the absorber inertia.

The success of the vibration absorber depends on several factors. First the harmonic excitation must not deviate much from its constant frequency ω . If the driving frequency

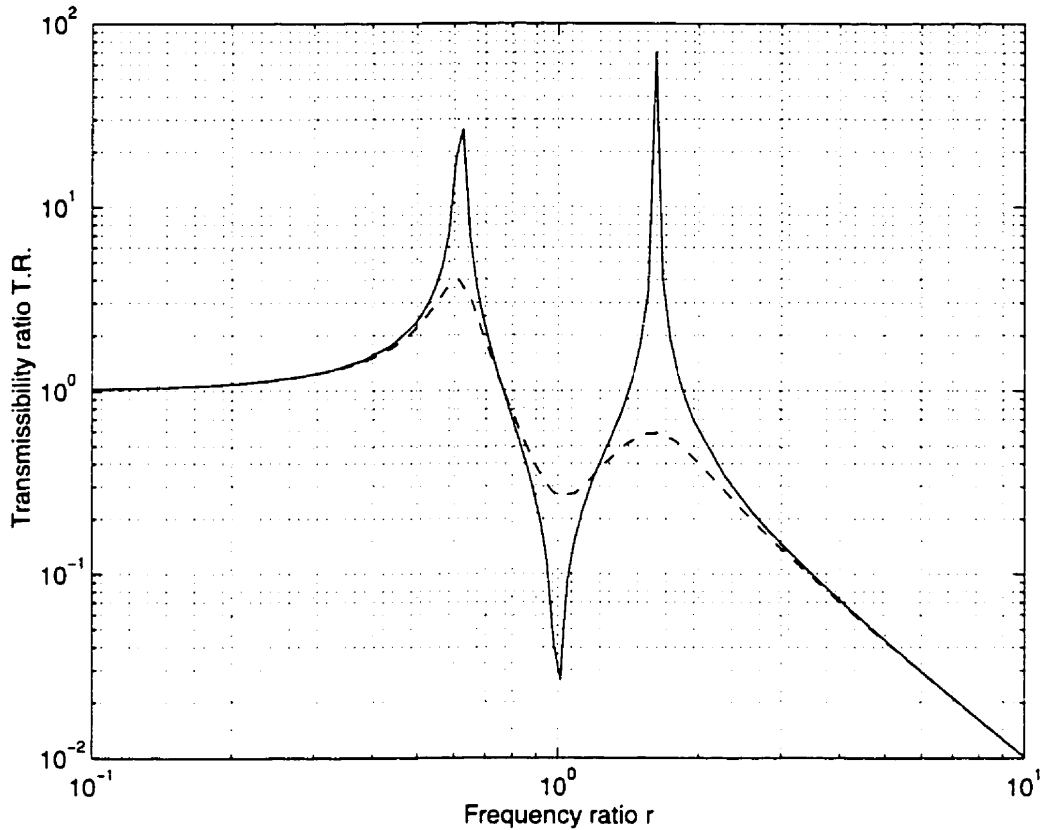


FIGURE 6.20. Plot of the transmissibility ratio in logarithmic scale for vibration absorber with (dashed) and without damping (solid)

drifts much, the tuning condition will not be satisfied, and the load will experience some oscillation. As illustrated in Figure 6.20, if damping is introduced to the vibration absorber, the amplitude of vibration of the load will not be zero, but the vibration absorption will be effective in a wider frequency range. For harmonic drive systems, the frequency of the torque ripple depends directly on the DC motor speed as explained in Chapter 3. Therefore, it is desired to have some damping in the vibration absorber, in order to make the absorption system more robust to variation of the operational speed of the harmonic drive. The general guideline for choosing the best value of the damping, can be find in [66]. Meanwhile, vibration absorber can only compensate for one harmonic of the torque ripple, hence, special care should be taken in the design so that the other dominant harmonics of the torque ripple does not coincide with the resonance frequency of the two-degree-of-freedom system. Another key factor in absorber design is that, according to Equation 6.19, the absorber spring must be capable of withstanding the full force of the excitation and hence must be capable of the corresponding deflections. The issue of spring size and deflection

as well as the value of the absorber inertia places a geometric limitation on the design of a vibration absorber system.

7. Summary

In this chapter robust $\mathcal{H}_\infty$ -based torque controllers are designed and implemented for harmonic drive systems under constrained- and free-motion applications. It is illustrated how effectively an empirical nominal model for the system is obtained through experimental frequency response estimates. By this means not only a linear model is proposed for the nominal model of the system, but also the deviation of the nonlinear system from the nominal model is encapsulated in a model uncertainty. This representation provides sufficient information to build a robust $\mathcal{H}_\infty$ -based torque controller for the harmonic drive system. Solving the mixed-sensitivity problem for a tracking and disturbance attenuation objective, an $\mathcal{H}_\infty$ controller is designed accommodating the actuator saturation limits. An integral part of torque feedback is the Kalman filtered torque of the intelligent built-in torque sensor. Implementing the controllers for two different setups under two different operating conditions, the performance of the closed-loop system is evaluated experimentally. Exceptional performance results are obtained from the time and frequency response of the closed-loop system, especially for constrained-motion application. To further improve the performance of the system for the free-motion case, a model-based friction-compensation algorithm is implemented. It is shown that compensation of estimated Coulomb and viscous friction reduces the system frequency response variations, and hence, model uncertainty. The uncertainty measure is therefore, not only used for control synthesis, but also as a quantitative indicator of the effectiveness of the friction-compensation algorithm. By comparison of the frequency and time domain performance of the system with and without friction-compensation, it is concluded that friction-compensation improves the performance of the system improved for tracking signals with low-frequency content. Finally, this chapter is concluded by presenting different methods to cancel torque ripples from the output torque.

CHAPTER 7

Conclusions

1. Synopsis

This thesis is devoted to the study of the robust torque control of harmonic drive systems. This section provides concluding remarks on three topics: torque sensing, modelling and simulation, and robust torque control of harmonic drive systems.

1.1. Torque Sensing

In this thesis an intelligent built-in torque sensing technique was developed in order to measure the transmitted torque. Sensor strain-gauges are mounted directly on the flexspline, and therefore, no extra flexible element is introduced into the system. To have maximum sensing accuracy, four Rosette strain gauges are employed using an accurate positioning method. An accurate drawing of the strain-gauge placement positions is printed on a transparent film, and the strain-gauges are placed on the transparent film using a microscope. It is shown that by accurately placing the transparent film on the flexspline, and cementing the strain-gauges onto the surface, the positioning error can be reduced to a minimum. Calibration of the torque sensor shows that the sensor performed linearly and the torque readings are not dependent to the position of the flexspline.

One important characteristic of harmonic drive torque transmissions, as observed in free motion experiments, is a high frequency oscillation in the output torque signal. To cancel these “torque ripples” from the measured torque, Kalman filter estimation is employed. Due to the dependence of the frequency content of the torque ripples on the wave generator velocity, a simple forth order harmonic oscillator proved sufficient to model the torque ripples. The performance of Kalman filter to cancel the torque ripples from torque

measurements is shown to be very fast and accurate. Moreover, the error model is extended to incorporate any misalignment torque signature. By on-line implementation of the Kalman filter, it is shown that this method is a fast and accurate way to filter torque ripples and misalignment torque, and hence, this intelligent built-in torque sensor is used as an integral part of the torque feedback loop.

1.2. Modelling and Simulation

Based on experimental and theoretical studies, a systematic way to capture and rationalize the dynamics of the harmonic drive systems was introduced in this thesis. Simple and accurate models for compliance, hysteresis and friction are established and model parameters are identified using least-squares approximation. A measure of variation is defined, by which the reliability of the estimated parameter for different operating condition, as well as the accuracy of the simple model is quantified. From compliance modelling results, it is concluded that identifying stiffness and structural damping together resolves the reported difficulties in determining the compliance parameters. Moreover, it is shown that a linear stiffness model best captures the behavior of system when combined with a good model for hysteresis. A simple static model for hysteresis is also introduced, and it is shown that this simple model can replicate the hysteresis effect in harmonic drives better than some other more complex dynamic models reported in the literature. Friction losses of the harmonic drive are modelled at both low and high velocities. From experiments on two different harmonic drives it is concluded that there is no stiction in the transmission, but rather a rising friction at low velocities.

The model performance is assessed by a simulation verifying the experimental results for both constrained- and free-motion cases. The simulated velocity and torque of the system is compared to experimental results for several constrained- and free-motion experiments. An accurate match is obtained for different operating ranges, which indicated the fidelity of the model to accurately replicate the dynamic behavior of the system, and confirms the effectiveness of modelling and parameter identification schemes to capture the dynamics of the harmonic drive systems. Moreover, the significance of individual components of the nonlinear model is assessed by a parameter sensitivity study. By comparing the simulation results of the complete model and a series of simplified models to the experiment results, it is concluded that accurate estimation of Coulomb and viscous friction significantly contributes to the accuracy of the simulations. Furthermore, it is shown that both stiffness and structural damping are prominent for the accuracy of the constrained-motion simulations.

1.3. Robust Torque Control

In this thesis robust $\mathcal{H}_\infty$ -based torque controllers are designed and implemented for harmonic drive systems under constrained- and free-motion applications. It is illustrated how effectively an empirical nominal model for the system is obtained through experimental frequency response estimates. By this means not only a linear model is proposed for the nominal model of the system, but also the deviation of the nonlinear system from the nominal model is encapsulated in a model uncertainty. This representation provides sufficient information to build a robust $\mathcal{H}_\infty$ -based torque controller for the harmonic drive system. Solving the mixed-sensitivity problem for a tracking and disturbance attenuation objective, an $\mathcal{H}_\infty$ controller is designed accommodating the actuator saturation limits. An integral part of torque feedback is the Kalman filtered torque of the intelligent built-in torque sensor. Implementing the controllers for two different setups under two different operating conditions, the performance of the closed-loop system is evaluated experimentally. Exceptional performance results are obtained from the time and frequency response of the closed-loop system, especially for constrained-motion application. To further improve the performance of the system for the free-motion case, a model-based friction-compensation algorithm is implemented. It is shown that compensation of estimated Coulomb and viscous friction reduces the system frequency response variations, and hence, model uncertainty. The uncertainty measure is therefore, not only used for control synthesis, but also as a quantitative indicator of the effectiveness of the friction-compensation algorithm. Finally, by comparison of the frequency and time domain performance of the system with and without friction-compensation, it is concluded that friction-compensation improves the performance of the system at low frequencies.

2. Directions For Future Research

We have developed an intelligent built-in torque sensor for harmonic drive systems. Although the performance of the sensor for torque feedback is excellent, more research may be required in order to use this torque sensor in industrial applications. We suggest further research for a more robust and rugged design of the sensor as well as further developments for a more effective on-board implementation of the Kalman filter estimation technique.

We also proposed robust $\mathcal{H}_\infty$ -based torque controllers for harmonic drive systems for constrained- and free-motion applications, and implemented a model-based friction compensation routine in free-motion applications. Moreover, a complete model of the harmonic

drive was developed for the purpose of model-based control. A suggested direction for future research is to employ the knowledge obtained through the modelling and identification of the harmonic drives into further development of other model-based control schemes. The effect of hysteresis in constrained-motion applications can be minimized through a model-based control algorithm which compensates for the hysteresis. Moreover, for low-velocity and fine-positioning applications where the effect of Stribeck friction is prominent, the model-based friction-compensation algorithm may be extended to incorporate Stribeck friction as well.

Another direction for future research is the implementation of the torque ripple compensation routine on the experimental setups. As elaborated in § 6 of Chapter 6, torque ripples are real high frequency torque oscillation observed in the output of the harmonic drive systems. In the applications where the smoothness of the output torque is important, it is desirable to compensate for the torque ripples. Further research is necessary in order to design a vibration isolator or absorber, in addition to incorporating a powerful amplifier in the testing station for the implementation of a torque ripple cancellation routine.

A natural extension of this research is to incorporate the torque controller designed for the harmonic drive systems for a multiple joint robot. Further research is necessary to design the supervisory control for the entire robotic manipulator. Experimental implementation of the overall control of a robotic manipulator can better show the merits of the developed torque control schemes in this thesis.

Finally, the intelligent built-in torque sensor, and the torque-controlled harmonic drive system have the potential for many industrial applications. Further research is therefore, necessary to adapt the design and implementation of the proposed techniques given in this thesis to accommodate the requirement for other applications.

REFERENCES

- [1] J. Adams and S. Payandeh. Methods for low-velocity friction compensation: theory and experimental study. *Journal of Robotic Systems*, 13(6):391–404, June 1996.
- [2] N.A. Aliev. A study of the dynamic behavior of flexible gears in harmonic drives. *Soviet Engineering Research*, 66(6):7–11, 1986.
- [3] B. Armstrong. Friction: Experimental determination, modelling and compensation. *Proceeding of IEEE International Conference on Robotics and Automation*, pages 1422–1427, 1988.
- [4] B. Armstrong. On finding exciting trajectories for identification experiments involving systems with nonlinear dynamics. *The International Journal of Robotics Research*, 8(6):28–48, December 1989.
- [5] B. Armstrong. Control of machines with non-linear, low-velocity friction: A dimensional analysis. *Experimental Robotics I*, pages 180–195, 1990.
- [6] B. Armstrong. Stick-slip arising from Stribeck friction. *Proceeding of IEEE International Conference on Robotics and Automation*, pages 1377–1382, 1990.
- [7] B. Armstrong-Helouvry. Friction lag and stick-slip. *Proceeding of IEEE International Conference on Robotics and Automation*, 2:1448–1453, 1992.
- [8] B. Armstrong-Helouvry. Stick slip and control in low speed motion. *IEEE Transactions on Automatic Control*, 38(10):1483–1496, October 1993.
- [9] B. Armstrong-Helouvry, P. Dupont, and C. Canudas de wit. A survey of models, analysis tools and compensation methods for control of machines with friction. *Automatica*, 30(7):1083–1138, 1994.
- [10] G.J. Balas, J.C. Doyle, K. Glover, A. Packard, and R. Smith. *μ -Analysis and Synthesis Toolbox*. The MathWorks, Inc., P.O. Box 1337, Minneapolis, MN 55414-5377, 1991.

- [11] P.R. Belanger. Estimation of angular velocity and acceleration from shaft encoder measurements. Technical Report TR-CIM-91-1, Centre for Intelligent Machines, McGill University, 1991.
- [12] P.R. Belanger. *Control engineering, a modern approach, chapter 8*. Saunders College Publishing, 6277 Sea Harbor Drive, Orlando, Florida 32887-6777, 1995.
- [13] A. Ben-Israel. *Generalized Inverses: Theory and Application*. Wiley, New York, 1974.
- [14] J.E. Bernard. The simulation of coulomb friction in mechanical systems. *Simulation*, 34(1):11-16, 1980.
- [15] P.J. Bliman. Mathematical study of Dahl's friction model. *European Journal of Mechanics, A / Solids*, 11(6):835-848, 1992.
- [16] B. Bona and M. Indri. Friction compensation and robust hybrid control. *Proceeding of IEEE International Conference on Robotics and Automation*, 2:81-86, 1993.
- [17] Stephan P. Boyd and Craig H. Barratt. *Linear Controller Design: Limits of Performance*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 07632, 1991.
- [18] M.M. Bridges, D.M. Dawson, and S.C. Martindale. Experimental study of flexible joint robots with harmonic drive gearing. *Proceedings of the IEEE Conference on Control Applications*, 2:499-504, 1993.
- [19] L. Cai and G. Song. A smooth robust nonlinear controller for robot manipulators with joint stick slip friction. *Proceeding of IEEE International Conference on Robotics and Automation*, 449-454 1993.
- [20] A. De Carli, S. Cong, and D. Matacchioni. Dynamic friction compensation in servodrives. *Proceedings of the IEEE Conference on Control Applications*, 1:193-198, 1994.
- [21] J.D. Chapel. Attaining impedance control objectives using $\mathcal{H}_\infty$ design methods. *Proceeding of IEEE International Conference on Robotics and Automation*, 2:1482-87, 1991.
- [22] J.H. Charlson. Harmonic drives for servomechanisms. *Machine design*, 57(1):102-106, January 1985.
- [23] P. Chedmail and J.P. Martineau. Characterization of the friction parameters of harmonic drive actuators. *International Conference on Dynamics and Control of Structures in Space*, 1:567-581, 1996.

- [24] J. Chiba. Adaptive control cuts position servo vibration. *Electrotechnology*, 6(1):17–19, 1995.
- [25] Reliance Motion Control. *Electro-Craft Servo Systems*. Eden Prairie, MN, 55344, 1993.
- [26] GMT International Corporation. *Vibration Isolation mounts*. GMT International Corporation, Villa Rica, Georgia, 30180, USA, 1994.
- [27] VME Microsystems International Corporation. *VMIVME-4116 Instructional Manual*. Huntsville. AL. 35803. 1987.
- [28] Leonardo da Vinci. The notebooks. Dover, N.Y., 1519.
- [29] P.R. Dahl. Solid friction damping of mechanical vibration. *AIAA Journal*, 14(12):1675–1682, December 1976.
- [30] C. Canudas de Wit. Experimental results on adaptive friction compensation in robot manipulators: Low velocities. *Experimental Robotics I*, pages 196–214, 1990.
- [31] C. Canudas de Wit, P Noel, A. Aubin, and B. Brogliato. Adaptive friction compensation in robot manipulators: Low velocities. *The International Journal of Robotics Research*, 10(3):189–199, June 1991.
- [32] C. Canudas de Wit and V. Seront. Robust adaptive friction compensation. *Proceeding of IEEE International Conference on Robotics and Automation*, pages 1383–1388, 1990.
- [33] M.E. Dohring, E. Lee, and W.S. Newman. A load-dependent transmission friction model: Theory and experiments. *Proceeding of IEEE International Conference on Robotics and Automation*, pages 430–436, 1993.
- [34] K.L. Doty, C. Melchiorri, and C. Bonivento. Theory of generalized inverses applied to robotics. *International Journal of Robotics Research*, 12(1):1–19, Feb 1993.
- [35] J. Doyle, K. Glover, P.P. Khargonekar, and B. A. Francis. State-space solution to standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems. *IEEE Transactions on Automatic Control*, AC-34(8):831–847, 1989.
- [36] J.C. Doyle, B.A. Francis, and A.R. Tannenbaum. *Feedback control theory*. Macmillan, N.Y., 1990.
- [37] J.C. Doyle and G. Stein. Multivariable feedback design: concepts for a classical modern synthesis. *IEEE Transactions on Automatic Control*, AC-26(4-16), 1981.

- [38] P.E. Dupont. Friction modeling in dynamic robot simulation. *Proceeding of IEEE International Conference on Robotics and Automation*, pages 1370–1376, 1990.
- [39] P.E. Dupont. The effect of coulomb friction on the existence and uniqueness of the forward dynamics problem. *Proceeding of IEEE International Conference on Robotics and Automation*, 2:1442–1447, 1992.
- [40] B.A. Francis and G. Zames. On $\mathcal{H}_\infty$ optimal sensitivity for SISO feedback system. *IEEE Transactions on Automatic Control*, AC-29:9–16, 1984.
- [41] K. Glover and J. Doyle. State-space formulae for all stabilizing controllers that satisfy an $\mathcal{H}_\infty$ norm bound and relation to risk sensitivity. *Systems and Control Letters*, 11:167–172, 1988.
- [42] I. Godler, K. Ohnishi, and T. Yamashita. Repetitive control to reduce speed ripple caused by strain wave gearing. *IECON Proceedings*, 2:1034–38, 1994.
- [43] A. Gogoussis and M. Donath. Coulomb friction effects on the dynamics of bearing transmissions in precision robot mechanisms. *Proceeding of IEEE International Conference on Robotics and Automation*, pages 1440–1446, 1988.
- [44] G.H. Golub and C. Van Loan. *Matrix computations*. The Johns Hopkins University Press, Baltimore, MD, 1989.
- [45] S.C.P. Gomes and J.P. Chretien. Dynamic modelling and friction compensated control of a robot manipulator joint. *Proceeding of IEEE International Conference on Robotics and Automation*, 2:1429–1435, 1992.
- [46] Michael Green and David J.N. Limebeer. *Linear Robust Control*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 07632, 1995.
- [47] Signal Analysis Group. *DSP Technology Inc. Siglab Version 1.0*. DSP Technology Inc., 48500 Kato Road, Fremont, CA 94538-7385, 1994.
- [48] D.A. Haessig and B. Friedland. On the modeling and simulation of friction. *Journal of Dynamic Systems, Measurement and Control*, 113(3):354–362, 1991.
- [49] M. Hashimoto. Robot motion control based on joint torque sensing. *Proceeding of IEEE International Conference on Robotics and Automation*, pages 256–261, 1989.
- [50] M. Hashimoto, T. Horiuchi, Y. Kiyosawa, and H. Hirabayashi. The effects of joint flexibility on robot motion control based on joint torque positive feedback. *Proceeding of IEEE International Conference on Robotics and Automation*, 2:1220–1225, April 1991.

- [51] M. Hashimoto, Y. Kiyosawa, H. Hirabayashi, and R.P. Paul. A joint torque sensing technique for robots with harmonic drives. *Proceeding of IEEE International Conference on Robotics and Automation*, 2:1034–1039, April 1991.
- [52] M. Hashimoto, Y. Kiyosawa, and R.P. Paul. A torque sensing technique for robots with harmonic drives. *IEEE Transaction on Robotics and Automation*, 9(1):108–116, Feb 1993.
- [53] M. Hashimoto, K. Koreyeda, T. Shimono, H. Tanaka, Y. Kiyosawa, and H. Hirabayashi. Experimental study on torque control using harmonic drive built-in torque sensors. *Proceeding of IEEE International Conference on Robotics and Automation*, pages 2026–2031, May 1992.
- [54] M. Hashimoto and R.P. Paul. Integration of manipulator joint sensor information for robust robot control. *Proceeding of IEEE Conference on Decision and Control*, pages 603–604, 1987.
- [55] D.P. Hess and A. Soom. Friction at a lubricated line contact operating at oscillating sliding velocities. *Journal of Tribology*, 112(1):147–152, January 1990.
- [56] N. Hogan. Impedance control of robots with harmonic drive system. *Proceeding of the American Control Conference*, 1:398–402, 1991.
- [57] L. Hsia. The analysis and design of harmonic gear drives. *Proceedings of the 1988 IEEE International Conference on Systems, Man and Cybernetics*, 1:616–619, 1988.
- [58] R. Hui, N. Kircanski, A. Goldenberg, C. Zhou, P. Kuzan, J. Weircienski, D. Gershon, and P. Sinha. Design of the iris facility - a modular, reconfigurable and expandable robot test bed. *Proceeding of IEEE International Conference on Robotics and Automation*, 1:155–160, 1993.
- [59] Data Translation Inc. *User Manual for DT1401-Series*. Marlboro, Ma, 01752, 1988.
- [60] HD Systems Inc. *Harmonic drive gearing*. Manufacturer's Cataloge, 1991.
- [61] HD Systems Inc. *Harmonic Drive Gearing: Cup Type HDUC and HIUC Component Sets*. Manufacturer's Cataloge, 1991.
- [62] HD Systems Inc. *Harmonic Drive Gearing: Precision DC Servo Actuators*. Manufacturer's Cataloge, 1991.
- [63] HD Systems Inc. *Harmonic Drive Gearing: Precision Gearheads Close Couple Model*. Manufacturer's Cataloge, 1991.

- [64] HD Systems Inc. *The World of Harmonic Drive Gearing*. Manufacturer's Cataloge, 1991.
- [65] Sky Computer Inc. *Challenger-C30/V Hardware Reference Manual*. CH30-VME-HR-90-1.1, 1990.
- [66] Daniel J. Inman. *Engineering Vibration, chapter 5, pp 240-264*. Prentice Hall, Englewood cliffs, N.J., 1996.
- [67] M.N. Ivanov. *Harmonic Gear Transmissions*. Visshaya shkola, M., (Russian), 1981.
- [68] K.S. Jeong, D.G. Lee, and S.H. Oh. Development of the composite flexspline for a cycloid-type harmonic drive using net shape manufacturing method. *Composite Structures*, 32(1-4):557-565, 1995.
- [69] K. Kaneko, T. Murakami, K. Ohnishi, and K. Komoriya. Torque control with non-linear compensation for harmonic drive DC motors. *IECON Proceedings*, 2:1022-1027, 1994.
- [70] D. Karnopp. Computer simulation of stick-slip friction in mechanical dynamic systems. *Journal of Dynamic Systems, Measurement and Control*, 107(1):100-103, 1985.
- [71] S. Kato, N. Sato, and T. Matsubayashi. Some consideration on characteristics of static friction of machine tool slideway. *Journal of Lubrication Technology*, 94(3):234-247, July 1972.
- [72] H. Kazerooni. Instrumented harmonic drives for robotic compliant maneuvers. *Proceeding of IEEE International Conference on Robotics and Automation*, 3:2274-79, 1991.
- [73] H. Kazerooni. Dynamics and control of instrumented harmonic drives. *Journal of Dynamic Systems Measurement and Control - Transactions of the ASME*, 117(1):15-19, March 1995.
- [74] P. Khargonekar and A. Tannenbaum. Noneuclidean metrics and the robust stabilization systems with parameter uncertainty. *IEEE Transactions on Automatic Control*, AC-30:1005-1013, 1985.
- [75] H. Kimura. Robust stabilization for a class of transfer functions. *IEEE Transactions on Automatic Control*, AC-29:788-793, 1984.
- [76] N. Kircanski, A. Goldenberg, and S. Jia. An experimental study of nonlinear stiffness, hysteresis, and friction effects in robot joint with harmonic drives and torque

- sensors. *Proceedings of the Third International Symposium on Experimental Robotics*, 1:147–154, 1993.
- [77] Y. Kiyosawa and M. Sasahara. Performance of a strain wave gearing using a new tooth profile. *International Power transmission and Gearing Conference*, 2:607–612, April 1989.
 - [78] T. Kubo, G. Anwar, and M. Tomizuka. Application of nonlinear friction compensation to robot arm control. *Proceeding of IEEE International Conference on Robotics and Automation*, 1:722–727, 1986.
 - [79] G. Iagnani and R. Faglia. Harmonic drive transmission: the effect of their elasticity, clearance and irregularity on the dynamic behaviour of an actual scara robot. *Robotica*, 10:369–375, October 1992.
 - [80] Aerospace Products DIV. Lord Corporation. *Vibration Isolation Products*. Lord Corporation, Erie, PA, USA, 1996.
 - [81] J.W. Macki, P. Nistri, and P. Zecca. Mathematical models for hysteresis. *SIAM Review*, 35(1):94–123, March 1993.
 - [82] M.V. Margulis and D.P. Volkov. Calculation of the torsional rigidity of a harmonic power drive with a disc generator. *Soviet Engineering Research*, 67(6):17–20, 1987.
 - [83] T. Marilier and J.A. Richard. Non-linear mechanic and electric behavior of a robot axis with a harmonic-drive gear. *Robotics and Computer-Integrated Manufacturing*, 5(2/3):129–136, 1989.
 - [84] I.D. Mayergoyz and G. Friedman. Generalized preisach model of hysteresis. *IEEE Transactions on Magnetics*, pages 212–217, 1989.
 - [85] I.D. Mayergoyz, G. Friedman, and C. Salling. Comparison of the classical and generalized preisach hysteresis models with experiments. *IEEE Transactions on Magnetics*, pages 3925–3927, 1989.
 - [86] D.C McFarlane and K. Glover. *Robust Controller design using normalized coprime plant descriptions*. vol. 138 in lecture notes in Control and Information Sciences, Springer-Verlag, N.Y., 1990.
 - [87] Kollmorgen Inland Motor. *Brushless DC Motors and Servo Amplifiers*. Radford, VI, 24141, 1990.
 - [88] Kollmorgen Inland Motor. *FAST Drive User's Manual*. Radford, VA, 24141, 1995.

- [89] S. Nicosia and P. Tomei. On the feedback linearization of robots with elastic joints. *Proceeding of IEEE Conference on Decision and Control*, 1:180–185, 1988.
- [90] K. Ogata. *Discrete - time control systems*. Prentice Hall, Englewood cliffs, N.J., 1995.
- [91] F. Ossart and G. Meunier. Comparison between various hysteresis models and experimental data. *IEEE Transactions on Magnetics*, pages 2837–2839, 1990.
- [92] James G. Owen. *Performance optimization of highly uncertain systems in $\mathcal{H}_\infty$* . PhD thesis. McGill University, 1993.
- [93] J.G. Owen and G. Zames. Robust $\mathcal{H}_\infty$ disturbance minimization by duality. *Systems & Control Letters*, 19(4):255–263, 1992.
- [94] J.G. Owen and G. Zames. Duality theory for MIMO robust disturbance rejection. *IEEE Transactions on Automatic Control*. 38(5):743–752, 1993.
- [95] J.G. Owen and G. Zames. Duality theory of robust disturbance attenuation. *Automatica*, 29(3):695–705, 1993.
- [96] H. Schempt and D.R. Yoerger. Study of dominant performance characteristics in robot transmissions. *Journal of Mechanical Design*, 115(3):472–482, Sep 1993.
- [97] W. Seyfferth, A.J. Maghzal, and J. Angeles. Nonlinear modeling and parameter identification of harmonic drive robotic transmissions. *Proceeding of IEEE International Conference on Robotics and Automation*, 3:3027–3032, 1995.
- [98] S.A. Shuvalov. Calculation of forces acting on members of a harmonic gear drive. *Russian Engineering Journal*, 59(10):5–9, 1979.
- [99] M.W. Spong. Adaptive control of flexible joint manipulators. *Systems & Control Letters*, 13(1):15–21, Jul 1989.
- [100] M.W. Spong, J.Y. Hung, S. Bortoff, and F. Ghorbel. Comparison of feedback linearization and singular perturbation techniques for the control of flexible joint robots. *Proceedings of the American Control Conference*, 1:25–30, 1989.
- [101] D. Stokic and M. Vukobratovic. Historical perspectives and state of the art in joint force sensory feedback control of manipulation robots. *Robotica*, 11(2):149–157, Mar-Apr 1993.
- [102] H.D. Taghirad. On the modelling and identification of harmonic drive system. Technical Report TR-CIM-97-02, Centre for Intelligent Machines, McGill University, January 1997.

- [103] H.D. Taghirad and P.R. Belanger. An experimental study on modelling and identification of harmonic drive systems. *Proceeding of IEEE Conference on Decision and Control*, 4:4725–30, Dec 1996.
- [104] H.D. Taghirad and P.R. Belanger. Robust torque control of harmonic drive systems. *Submitted to the IEEE Transactions on Robotics and Automation*, Nov 1996.
- [105] H.D. Taghirad and P.R. Belanger. Intelligent torque sensing and robust torque control of harmonic drive under free-motion. *Proceeding of IEEE International Conference on Robotics and Automation*, 2:1749–54, April 1997.
- [106] H.D. Taghirad and P.R. Belanger. Modelling and parameter identification of harmonic drive systems. *Accepted for Publication in the Journal of Dynamic Systems, Measurements, and Control*, ASME, 1997.
- [107] H.D. Taghirad and P.R. Belanger. Robust torque control of harmonic drive under constrained motion. *Proceeding of IEEE International Conference on Robotics and Automation*, 1:248–253, April 1997.
- [108] H.D. Taghirad, P.R. Belanger, and A. Helmy. An experimental study on harmonic drive. *Technical Report Submitted to the International Submarine Engineering Ltd., Port Coquitlam, BC, Canada*, <http://www.cim.mcgill.ca/~taghirad>, 1996.
- [109] H.D. Taghirad, A. Helmy, and P.R. Belanger. Intelligent built-in torque sensor for harmonic drive system. *Submitted to the IEEE Transactions on Instrumentation and Measurement*, Dec 1996.
- [110] H.D. Taghirad, A. Helmy, and P.R. Belanger. Intelligent built-in torque sensor for harmonic drive system. *Proceedings of the 1997 IEEE Instrumentation and Measurement Technology Conference*, 1:969–974, May 1997.
- [111] Harmonic Drive Technologies. *Harmonic Drive Gearing: HDC Component Gear Set Selection Guide*. Manufacturer's Cataloge, 1993.
- [112] Harmonic Drive Technologies. *Harmonic Drive Gearing: Pancake Component Gear Sets*. Manufacturer's Cataloge, 1993.
- [113] Harmonic Drive Technologies. *Power transmission Products*. Manufacturer's Cataloge, 1993.
- [114] Y. Terauchi, K. Nagamura, and K. Ikejo. Study on friction loss of internal gear drives. *JSME International Journal, Series 3: Vibration. Control Engineering, Engineering for Industry*, 34(1):106–112, 1991.

- [115] Bruce R. Thompson. A planar harmonic drive robot for joint torque control. *MIT Master Thesis*, July 1990.
- [116] D.C. Threlfall. The inclusion of coulomb friction in mechanisms programs with particular reference to DRAM. *Mech. and Mach. Theory*, 13:475–483, 1978.
- [117] A. Tustin. The effects of backlash and speed-dependent friction on the stability of closed-cycle control systems. *J. Inst. Elec. Eng.*, 94(2A):143–151, 1947.
- [118] T.D. Tuttle. Understanding and modeling the behavior of a harmonic drive gear transmission. Technical Report 1365. MIT Artificial Intelligence Laboratory. 1992.
- [119] T.D. Tuttle and W. Seering. Kinematic error, compliance, and friction in a harmonic drive gear transmission. *Advances in Design Automation*, 65-1:319–324, 1993.
- [120] T.D. Tuttle and W. Seering. Modeling a harmonic drive gear transmission. *Proceeding of IEEE International Conference on Robotics and Automation*, 2:624–629, 1993.
- [121] T.D. Tuttle and W.P. Seering. A nonlinear model of a harmonic drive gear transmission. *IEEE Transaction on Robotics and Automation*, 12(3):368–374, June 1996.
- [122] M. Verma and E. Jonckheere. $\mathcal{L}_\infty$ compensation with mixed sensitivity as a broadband matching problem. *Systems and Control Letters*, 4:125–129, 1984.
- [123] D.P. Volkov. *Harmonic Drives*. Tekhnika, Kiev, (Russian), 1976.
- [124] D.P. Volkov and Y.N. Zubkov. Vibration in a drive with harmonic gear transmission. *Russian Engineering Journal*, 58(5):17–21, 1978.
- [125] C.D. Walrath. Adaptive bearing friction compensation based on recent knowledge of dynamic friction. *Automatica*, 20(6):717–727, 1984.
- [126] G. Zames. On the input-output stability of time-varying nonlinear feedback systems. *IEEE Transcript on Automatic Control*, II:228–238 (part I), and 465–476 (Part II), 1966.
- [127] G. Zames and B.A. Francis. Feedback, minimax sensitivity and optimal robustness. *IEEE Transactions on Automatic Control*, AC-28(5):585–601, 1983.
- [128] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 07632, 1996.

APPENDIX A

Mathematical Background

1. Least-Squares Estimation

1.1. Linear Regression Models

In order to estimate unknown parametric quantities from measurements we must assume model representations and express them in such a way that emphasizes the explicit relationship between the unknown quantities and the measurements. Many models are affine in unknown quantities (denoted θ) and can be expressed as:

$$\mathbf{A}\theta = \mathbf{b} + \mathbf{n} \quad (\text{A.1})$$

where $\mathbf{A}$ is an $m \times n$ matrix called the *observation matrix*, θ is an $n \times 1$ column vector of identification parameters called *parameter vector*, $\mathbf{b}$ is an $m \times 1$ column vector called the *measurement vector*, and $\mathbf{n}$ is an $m \times 1$ vector called the *measurement noise*, where $\mathbf{n}$ is assumed to be zero-mean. If there is more measurements than unknowns, i.e. $m > n$, Equation A.1 constitutes overdetermined system of equations. The extra measurements are used to offset the effects of the noise. The objective of least-squares estimation is to find an estimate of the model parameter, called $\hat{\theta}$, which approximately solves

$$\mathbf{A}\hat{\theta} \approx \mathbf{b} \quad (\text{A.2})$$

with minimum least-squares error. Equation A.2 is called a *linear regression model* and $\hat{\theta}$ is called the *least-squares solution*. If $\mathbf{A}$ is of full column rank, the least-squares solution can be expressed in terms of *Moore-Penrose generalized inverse* or *pseudo-inverse* of the rectangular matrix $\mathbf{A}$,

$$\hat{\theta} = \mathbf{A}^\dagger \mathbf{b} \quad (\text{A.3})$$

where

$$\mathbf{A}^\dagger := (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \quad (\text{A.4})$$

Direct evaluation of $\mathbf{A}^\dagger$ can be numerically intractable due to ill conditioning. For the purpose of this thesis an alternative method is used [44], which applies Householder reflections¹. With Householder reflections orthogonal triangular factorization is used to find the pseudo-inverse. Specifically the matrix $\mathbf{A}$ is factorized into

$$\mathbf{AP} = \mathbf{QR} \quad (\text{A.5})$$

where $\mathbf{P}_{n \times n}$ is a permutation, $\mathbf{Q}_{m \times n}$ is orthogonal and $\mathbf{R}_{n \times n}$ is upper triangular. The least-squares solution $\hat{\boldsymbol{\theta}}$ is computed from

$$\hat{\boldsymbol{\theta}} = \mathbf{PR}^{-1}(\mathbf{Q}^T \mathbf{b}) \quad (\text{A.6})$$

Where $\mathbf{R}^{-1}$ is calculated by back substitution.

1.2. Nonlinear Regression Models

If the relationship between the unknown parametric quantities and the measurements is not linear in $\boldsymbol{\theta}$, the model is called *nonlinear regression* and can be expressed as:

$$\mathbf{f}(\boldsymbol{\theta}) + \mathbf{n} = \mathbf{0} \quad (\text{A.7})$$

where $\mathbf{f}$ is an $m \times 1$ vector valued function of $\boldsymbol{\theta}$ and $\mathbf{n}$ is the measurement noise. Consistent with linear regression one objective is to find a least-squares solution $\hat{\boldsymbol{\theta}}$ which minimizes the norm of error,

$$e(\hat{\boldsymbol{\theta}}) = \frac{1}{2} \mathbf{f}^T \mathbf{f}. \quad (\text{A.8})$$

A solution can be typically computed iteratively, using the *Newton-Gauss* method². In this method from an initial guess of the parameter vector $\hat{\boldsymbol{\theta}}_0$, a sequence of $\hat{\boldsymbol{\theta}}_k$ is computed iteratively via

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k + \Delta \boldsymbol{\theta}_k \quad (\text{A.9})$$

where the least-squares solution of the system is,

$$\Delta \boldsymbol{\theta}_k = -(\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{f}(\boldsymbol{\theta}^k) \quad (\text{A.10})$$

$\mathbf{J}(\boldsymbol{\theta})$ is the $m \times n$ Jacobian matrix of the vector function $\mathbf{f}(\boldsymbol{\theta})$ defined by:

$$\mathbf{J}(\boldsymbol{\theta}) = \frac{\partial \mathbf{f}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \quad (\text{A.11})$$

¹Operator \ in Matlab

²Function *leastsq* in Matlab

A stopping criteria of $\|\Delta\theta_p\| < \epsilon$ is typically used, and the last element of sequence $\hat{\theta}_{p+1}$, is the least-squares solution sought. Direct calculation of the inverse in Equation A.10 can be numerically intractable due to ill conditioning. Therefore, other numerical approaches such as Householder reflection, or Cholesky decomposition is used in numerical computations.

2. $\mathcal{H}_\infty$ Controller Design

2.1. Norms for Signals and Systems

2.1.1. The Signal Spaces $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{H}_2$

There are many ways to describe the size of a signal. Among them, those that satisfy certain geometric properties are of interest. These measures are called *norms*, and the geometric properties that norms satisfy are expressed in the framework of a vector space. Suppose that $\mathcal{V}$ is a vector space of functions $v : \mathbb{R} \rightarrow \mathbb{R}$ and $\|\cdot\| : \mathcal{V} \rightarrow \mathbb{R}_+$. $\|\cdot\|$ is a norm on $\mathcal{V}$ if it satisfies the following properties:

- 1: $\|v\| \geq 0 \quad \forall v \in \mathcal{V}$
- 2: $\|v\| = 0 \Leftrightarrow v = 0$
- 3: $\|av\| = |a|\|v\|, \quad \forall a \in \mathbb{R}$
- 4: $\|u + v\| \leq \|u\| + \|v\|$

This notation emphasizes that a norm is a generalization of the absolute value for real or complex numbers, and the Euclidean length of a vector [17].

The $\mathcal{L}_1$ and $\mathcal{L}_2$ norm of signal $u(t)$ can be defined as

$$\|u\|_1 := \int_{-\infty}^{\infty} |u(t)| dt \quad (\text{A.12})$$

$$\|u\|_2 := \left(\int_{-\infty}^{\infty} u^*(t)u(t) dt \right)^{\frac{1}{2}} \quad (\text{A.13})$$

and $\mathcal{L}_1$ and $\mathcal{L}_2$ are the linear spaces of all signals, or vector signals, with bounded $\mathcal{L}_1$ and $\mathcal{L}_2$ norms respectively. Alternatively, $\mathcal{L}_2$ may be thought of as a frequency domain space, with norm

$$\|\bar{u}\|_2 = \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \bar{u}^*(jw)\bar{u}(jw) dw \right)^{\frac{1}{2}} \quad (\text{A.14})$$

where $\bar{u}$ is the Fourier transform of u . Parseval's theorem states that the $\mathcal{L}_2$ norm of the signal calculated in the time and frequency domains are equal: $\|\bar{u}\|_2 = \|u\|_2$. Hence, we

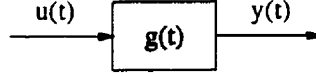


FIGURE A.1. General representation of a system and its input and output

consider these spaces as being one³ and the same and use the same notation for signals in the time and frequency domains. Thus, when we say that u is in $\mathcal{L}_2$ we mean that u is a signal of bounded energy, which can be thought of either as a function of time or frequency depending on the context [46].

In the time domain, $\mathcal{L}_2$ can be decomposed into a direct sum of $\mathcal{L}_{2+}$ and $\mathcal{L}_{2-}$, where $\mathcal{L}_{2+}$ is the space of signals defined for positive time, and zero otherwise. Similarly, $\mathcal{L}_{2-}$ is the space of signals defined for negative time, and zero otherwise. In the frequency domain, $\mathcal{L}_2$ can be similarly decomposed into $\mathcal{H}_2$ and $\mathcal{H}_2^\perp$, where Hardy space $\mathcal{H}_2$ is the space of Fourier transform of signals in $\mathcal{L}_{2+}$ and $\mathcal{H}_2^\perp$ is the space of Fourier transforms of signals in $\mathcal{L}_{2-}$. Thus, when we say that x is in Hardy space $\mathcal{H}_2$ we mean that x is a signal of bounded energy which, when considered in the time domain, is zero for negative time.

2.1.2. The System Spaces $\mathcal{L}_2$, $\mathcal{L}_\infty$ and $\mathcal{H}_\infty$

It is sufficient for the purpose of this thesis to consider only the input-output systems that are linear, time-invariant, and finite dimensional. For a single-input single-output (SISO) system in the time domain, an input-output model for such a system has the form of a convolution integral,

$$y(t) = \int_{-\infty}^{\infty} g(t - \tau)u(\tau)d\tau \quad (\text{A.15})$$

where $g(t)$ is the impulse response of the system as illustrated in Figure A.1. Let $G(s)$ denote the transfer function, the Laplace transform of $g(t)$. We say G is stable if it is analytic in the closed right half plane ($\Re(s) \geq 0$), and proper if $G(j\infty)$ is finite. For causal system, G is proper and $g(t) = 0$ for $t < 0$. G is rational with real coefficients for finite dimensional systems. The $\mathcal{L}_2$ and $\mathcal{L}_\infty$ norms for the transfer function G are defined by:

$$\|G\|_2 := \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} G^*(jw)G(jw)dw \right)^{\frac{1}{2}} \quad (\text{A.16})$$

$$\|G\|_\infty := \sup_w |G(jw)| \quad (\text{A.17})$$

³Actually equivalent classes of functions

and for stable G , by Parseval's theorem

$$\|G\|_2 := \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} G^*(j\omega)G(j\omega)d\omega \right)^{\frac{1}{2}} = \left(\int_{-\infty}^{\infty} g^*(t)g(t)dt \right)^{\frac{1}{2}}. \quad (\text{A.18})$$

$\mathcal{L}_2$ and $\mathcal{L}_\infty$ are the linear spaces of all systems, with bounded $\mathcal{L}_2$ and $\mathcal{L}_\infty$ norms. $\mathcal{L}_\infty$ denotes the space of all transfer functions which have no poles on the imaginary axis (i.e. with finite $\mathcal{L}_\infty$ norm). The Hardy space $\mathcal{H}_\infty$ is a subspace of $\mathcal{L}_\infty$ which considers only stable system, transfer function G with no poles in closed right half plane $\Re(s) \geq 0$.

To define system norms for multi-input multi-output (MIMO) systems, let us first define the Frobenious norm for an $m \times r$ matrix A ,

$$\|A\|_2 := \left(\sum_{i=1}^m \sum_{j=1}^r A_{ij}^2 \right)^{\frac{1}{2}} \quad (\text{A.19})$$

or

$$\|A\|_2^2 = \text{tr}(A^T A) = \text{tr}(A A^T) \quad (\text{A.20})$$

where “tr” refers to the trace, i.e., the sum of the diagonal elements. Alternatively the maximum singular value, $\bar{\sigma}(A)$ is defined as the maximum “gain” of the matrix which represents the *induced norm* of a matrix as:

$$\bar{\sigma}(A) := \sup_{u \neq 0} \frac{\|Au\|_2}{\|u\|_2} \quad (\text{A.21})$$

The maximum singular value can be calculated as $\bar{\lambda}^{\frac{1}{2}}(A^* A)$, where $\bar{\lambda}$ denotes the maximum eigenvalue of the matrix.

Linear, time-invariant MIMO systems are generalization of matrices [37]. By analogy to Frobenious, and induced norm, $\mathcal{L}_2$ and $\mathcal{L}_\infty$ norm for an $m \times r$ transfer matrix $G(s)$ can be defined respectively as [86, 128]:

$$\|G\|_2 := \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \text{tr}[G^*(j\omega)G(j\omega)]d\omega \right)^{\frac{1}{2}} \quad (\text{A.22})$$

$$\|G\|_\infty := \sup_{\omega} \bar{\sigma}[G(j\omega)]. \quad (\text{A.23})$$

2.2. Feedback Control Systems: Expressions and Notations

Figure A.2 shows the basic configuration for a feedback control system, where the controller C is placed in the feedforward path preceding the plant P . By use of feedback, the control input u is generated from the output error e , the difference between the desired output y_d and the output y . In the absence of the measurement noise, the transfer functions from the desired output y_d , and the disturbance d to the output y can be derived from [11]:

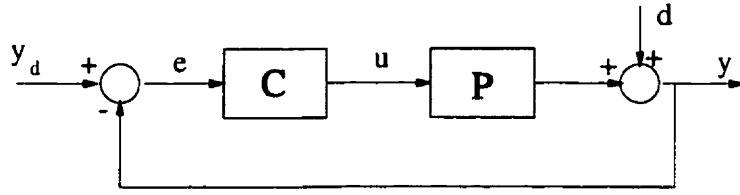


FIGURE A.2. Block diagram for feedback control system

$$y(s) = \frac{CP}{1+CP}y_d(s) + \frac{1}{1+CP}d(s) \quad (\text{A.24})$$

hence, the error can be represented as:

$$e(s) = y_d(s) - y(s) = \frac{1}{1+CP}y_d(s) - \frac{1}{1+CP}d(s) \quad (\text{A.25})$$

The transfer function $C(s)P(s)$ is called the *loop gain* and denoted by $L(s)$. For large loop gain

$$\frac{CP}{1+CP} \approx 1 \quad \text{and} \quad \frac{1}{1+CP} \approx \frac{1}{CP} \quad (\text{A.26})$$

High loop gain is therefore desirable at frequencies in the passband, since it makes y/y_d approximately equal to 1 (good tracking performance) and y/d small (disturbance rejection). The transfer function between the desired output y_d and the output y is called the *closed-loop transfer function* and can be represented by $T(s) = \frac{CP}{1+CP}$.

2.2.1. Sensitivity and Complementary Sensitivity

As was shown in the preceding section the transfer functions associated with a control system are implicit functions of the plant transfer function P . If $P(s)$ changes from $P_o(s)$ to $P_o(s) + \Delta P(s)$, then the closed-loop transfer function $T(s)$ changes from $T_o(s)$ to $T_o(s) + \Delta T(s)$. The *sensitivity* of $T(s)$ with respect to P is defined by the fractional change in $T(s)$ relative to the fractional change in $P(s)$, namely

$$S(s) = \frac{\Delta T(s)/T_o(s)}{\Delta P(s)/P_o(s)} \quad (\text{A.27})$$

For infinitesimal changes,

$$S(s) = \frac{\partial T(s)/T_o(s)}{\partial P(s)/P_o(s)} \quad (\text{A.28})$$

If a control system results in $|S(j\omega)| < 1$, then the sensitivity at frequency ω has been reduced by the control system, and the changes in the plant transfer function have less effect on the closed loop system performance.

For feedback control system illustrated in Figure A.2, $T(s) = \frac{C(s)P(s)}{1+C(s)P(s)}$ and the sensitivity can be calculated from Equation A.28 as:

$$S(s) = \frac{\partial T(s)/T(s)}{\partial P(s)/P(s)} = \frac{1}{1+CP} \quad (\text{A.29})$$

There is an algebraic relation between the closed-loop transfer function T and the sensitivity,

$$T(s) + S(s) = \frac{CP}{1+CP} + \frac{1}{1+CP} = 1 \quad (\text{A.30})$$

Therefore $T(s)$ is also called the *complementary sensitivity*. The Equations A.24 and A.25 can be rewritten in terms of sensitivity and complementary sensitivity as follows.

$$y(s) = T(s)y_d(s) + S(s)d(s) \quad (\text{A.31})$$

$$e(s) = S(s)y_d(s) - S(s)d(s) \quad (\text{A.32})$$

Thus, making S small in the passband simultaneously desensitizes the system, accommodates disturbance attenuation, and reduces tracking error, hence, constituting an objective in controller design.

2.2.2. Internal Stability

The internal stability of a the system can be assured by placing some conditions on the sensitivity and complementary sensitivity of the closed-loop system. Theorem A.1 summerizes these conditions [11]:

Theorem (A.1): The system of Figure A.2 is internally stable if, and only if, $T(s)$, $P^{-1}(s)T(s)$, and $P(s)S(s)$ are stable.

If $P(s)S(s)$ is to be stable, S must cancel out the right half-plane (RHP) poles of $P(s)$. If P has an unstable pole of multiplicity m at $s = p_o$, S must have a zero of the same multiplicity at the same point. It can be shown that this implies that

$$S(p_o) = \frac{dS}{ds}(p_o) = \cdots = \frac{d^{m-1}S}{ds^{m-1}}(p_o) = 0 \quad (\text{A.33})$$

Similarly, if $P^{-1}(s)T(s)$ is to be stable, T must cancel out the RHP zeros of P . If z_o is an RHP zero of P with multiplicity m , then

$$T(z_o) = \frac{dT}{ds}(z_o) = \cdots = \frac{d^{m-1}T}{ds^{m-1}}(z_o) = 0 \quad (\text{A.34})$$

Equations A.33 and A.34 are called *interpolation conditions* and show how the system dynamics constrain the admissibility of S and T .

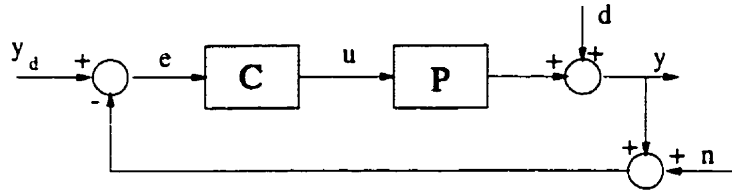


FIGURE A.3. Block diagram for feedback control system with measurement noise

2.2.3. Performance Limitations

Limitations on performance originate from actuators, sensors, stability conditions and robustness requirements. Figure A.3 restores the measurement noise, n . Equations A.31 and A.32 are valid here, but the effect of n is to be added. With all inputs, y_d , d , and n it can be shown that [11],

$$y(s) = T(s)y_d(s) + S(s)d(s) - T(s)n(s) \quad (\text{A.35})$$

$$e(s) = S(s)y_d(s) - S(s)d(s) + T(s)n(s) \quad (\text{A.36})$$

In the absence of sensor noise, it is possible to simultaneously desensitize the system, attenuate disturbance, and reduce tracking error by making $|S|$ small. In the presence of sensor noise however, making $|S|$ small constrains $T(s)$ to be close to one. This has the effect of transmitting the sensor noise directly to the output and the error.

If the sensor noise has high spectral content in the system passband, its contribution to the error may be dominant. Otherwise, it is still possible to achieve good performance by minimizing *weighted sensitivity* $\|W_s S\|$, where W_s shapes the sensitivity to be small only within the design bandwidth, while S can be large, and hence T can be small, for the frequencies where measurement noise has high spectral content.

To address the actuator effort and its imposed limitation on performance, from Figure A.3 the actuator effort u can be calculated from,

$$u(s) = P^{-1}(y - d) = P^{-1}T(y_d - d - n) \quad (\text{A.37})$$

Similar to the limitation imposed on T by measurement noise, if $T(j\omega) \approx 1$ at frequencies beyond the natural bandwidth of P , then $|P(j\omega)|$ will decrease well below 1 while $|P^{-1}(j\omega)T(j\omega)|$ will be large, and the actuator may not have enough bandwidth to provide the required power. Thus, it is desired not only to minimize a weighted sensitivity $W_s S$, to overcome the measurement noise limitation, but also to limit the weighted actuator effort $|W_u P^{-1}T|$, where W_u forces the actuator effort to be smaller than the allowable limit within the design bandwidth.

The performance limitation due to stability conditions is addressed in § 2.2.2. In the absence of RHP poles and zeros of $P(s)$, there are no restriction on S and T other than stability. From the interpolation conditions, $S = 0$ at RHP poles of P , and $S = 1$ at RHP zeros of P . RHP poles by themselves do not limit the ability to make $|S(j\omega)|$ small; however, RHP zeros do limit the ability to make $|S(j\omega)|$ small, because they force $S(s)$ to be 1 at some points in the complex plane. The consequence is that the overall sensitivity is increased everywhere. If P has a pole and zero close to each other in the right half-plane, it can be shown [36], that they can greatly amplify the constraints on S that either would induce alone. The performance limitations imposed by robustness requirements are due the non-ideal representation of the system by a mathematical model which possess uncertainty, which is addressed in § 2.4.

2.3. Uncertainty Representations

No mathematical representation can exactly model a physical system. For this reason the modelling errors might adversely affect the performance of a control system. The basic technique to represent the uncertainty is to model the plant as belonging to a set of systems $\mathcal{P}$. Such a set can be either *structured* or *unstructured*. One type of structured set is parameterized by a finite number of scalar parameters. *i.g.*, modelling an automobile suspension system as a mass, spring and dash-pot system where the exact values of the mass, the spring stiffness, and the dash-pot viscous damping, are only known with a relative error. Another type of structured uncertainty is represented by a discrete set of plants, not necessarily parameterized explicitly.

Unstructured uncertainties are more important because of two reasons. First, all models used in feedback design in the context of this thesis, should include some unstructured uncertainty to cover the unmodeled dynamics. Second, for a specific type of unstructured uncertainty, it is possible to develop simple and general analysis and synthesis methods. A simple and useful representation of unstructured set is that of a disk-like or multiplicative uncertainty.

2.3.1. Multiplicative Uncertainty

Suppose that the nominal plant transfer function is $P_o(s)$ and consider perturbed plant set as illustrated in Figure A.4 of the form

$$\forall P(s) \in \mathcal{P}, \quad P(s) = (1 + \Delta(s)W(s))P_o(s) \quad (\text{A.38})$$

where $W(s)$ is a fixed transfer function, called the *uncertainty weighting function* and Δ is a variable $\mathcal{H}_\infty$ transfer function of norm less than unity [36]. Furthermore, it is assumed

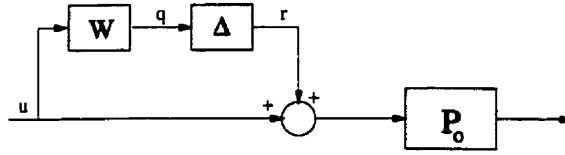


FIGURE A.4. Block diagram for multiplicative uncertainty representation

that the number of unstable poles of P_o and P are the same. Such a perturbation Δ is said to be *allowable*. Note that in this representation $\Delta(s)W(s)$ gives the normalized system variation away from 1 at each frequency:

$$\frac{P(j\omega)}{P_o(j\omega)} - 1 = \Delta(j\omega)W(j\omega) \quad (\text{A.39})$$

Now as $\|\Delta\|_\infty \leq 1$, we may conclude that

$$\left| \frac{P(j\omega)}{P_o(j\omega)} - 1 \right| \leq |W(j\omega)|, \quad \forall \omega \quad (\text{A.40})$$

This inequality describes a disk in the complex plane. At each frequency the point P/P_o lies in the disk with centre 1 and radius $|W(j\omega)|$. Typically, $|W(j\omega)|$ is an increasing function of ω : uncertainty increases with increasing frequency, since the effect of unmodeled dynamics are more vital at high frequencies. The main purpose of Δ is to account for the phase uncertainty and to act as a normalizing factor on the magnitude of the perturbation.

Thus this uncertainty model is characterized by a nominal plant P_o together with a weighting function W . How we obtained the weighting function in practice for the harmonic drive systems is addressed in § 1 of Chapter 6.

2.4. Robust Stability

Consider the plant transfer function P_o belonging to a set $\mathcal{P}$, as in the preceding section. A controller C illustrated in Figure A.5, provides *robust stability* if it provides

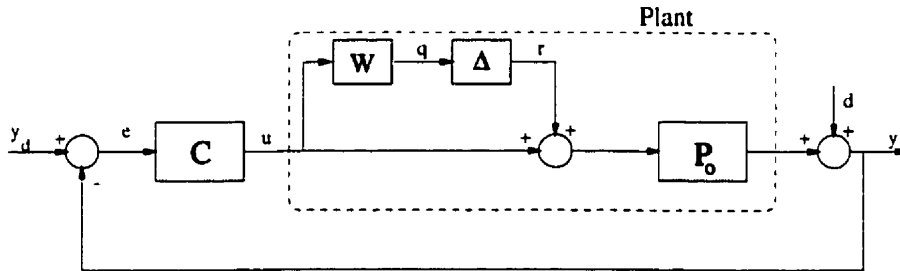


FIGURE A.5. Block diagram for feedback control system with multiplicative uncertainty for plant

internal stability not only for the nominal plant P_o , but also for every plant P in the set $\mathcal{P}$. To give conditions for robust stability let us first introduce the *small-gain theorem* [126]:

Theorem (A.2): Small-Gain Theorem

For the system in Figure A.6, where $G(s)$ is stable and strictly proper transfer function, the closed loop system is internally stable if $\|G\|_\infty \|\Delta\|_\infty < 1$.

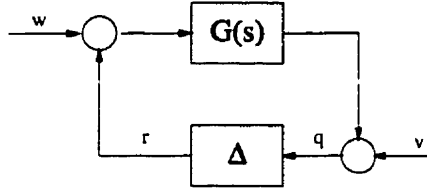


FIGURE A.6. Feedback system used in small-gain theorem

For systems with multiplicative uncertainty, Theorem A.3 applies directly to give a robust stability condition [36, 75, 122]:

Theorem (A.3): C provides robust stability for the system of Figure A.5 if, and only if, $\|WT\|_\infty < 1$.

There is a simple way to see the relevance of the condition $\|WT\|_\infty < 1$ to the robust stability using *small-gain theorem*. Consider the block diagram of the uncertain feedback system illustrated in Figure A.5, but ignore the inputs y_d and d . The transfer function from r , the output of Δ , to q , the input of Δ , equals $-WT$. Figure A.7 shows the modified block diagram of the system showing only the blocks Δ and $-WT$. Robust stability is guaranteed by having $\|\Delta\|_\infty \|WT\|_\infty < 1$, using small-gain theorem. But $\|\Delta\|_\infty < 1$ for all allowable perturbations, hence $\|WT\|_\infty < 1$ is the condition for robust stability [74].

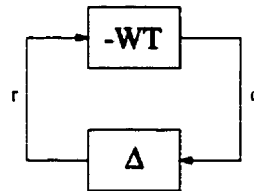


FIGURE A.7. Modified block diagram of system in Figure A.5 for robust stability analysis

2.5. Robust Performance

Similar to robust stability notion, robust performance can be defined as follows. Controller C provides *robust performance* if it provides performance not only for the nominal plant P_o , but also for every plant P in the set $\mathcal{P}$. The first performance index is robust stability, hence a system having robust performance should be robustly stabilized. As elaborated in § 2.2.3 the *nominal performance* condition (performance condition for nominal plant P_o) can be related to a weighted sensitivity norm by $\|W_s S\|_\infty < 1$. For multiplicative uncertainty P is perturbed to $(1 + \Delta W)P$, hence S is perturbed to

$$\frac{1}{1 + (1 + \Delta W)L} = \frac{S}{1 + \Delta WT} \quad (\text{A.41})$$

where L is the nominal loop gain, and T is the complementary sensitivity. Clearly, the *robust performance* condition should therefore be:

$$\|WT\|_\infty < 1 \quad \text{and} \quad \left\| \frac{W_s S}{1 + \Delta WT} \right\|_\infty < 1, \quad \forall \text{ allowable } \Delta \quad (\text{A.42})$$

Theorem (A.4) [36, 40, 127] represents the robust performance condition for a system with multiplicative uncertainty in terms of the function $|W_s S| + |WT|$.

Theorem (A.4): C provides robust performance for the system of Figure A.5 if, and only if, $\| |W_s S| + |WT| \|_\infty < 1$.

This condition is also called a *two-disk problem* as per the graphical interpretation of this theorem. For each frequency ω , construct two closed disks in the complex plane: one with center -1 and radius $|W_s(j\omega)|$; the other with center at the loop gain $L(j\omega)$ and radius $|W(j\omega)L(j\omega)|$. Then robust performance holds if, and only if, for each ω these two disks are disjoint.

The robust performance problem as stated in Theorem (A.4) was recently solved by Owen and Zames [92–95] using convex optimization. However, no software implementation was provided. The robust performance problem can be modified so that an approximate solution exists. A well known approximation introduced in the literature is the *mixed-sensitivity problem* which is the topic of the next section. We used the mixed-sensitivity problem in the context of this thesis for controller design, since there is a reliable software implemented in the form of Matlab toolboxes [10].

2.6. Mixed-Sensitivity Problem

A compromise to robust performance is to find a controller C such that the $\mathcal{H}_\infty$ norm of a transfer matrix consisting of weighted sensitivity and complementary sensitivity is smaller

than one, *i.e.*

$$\left\| \begin{bmatrix} WT \\ W_s S \end{bmatrix} \right\|_\infty < 1 \quad (\text{A.43})$$

This problem is called *mixed-sensitivity* in the literature and there is technical machinery available to solve this problem. It can be shown that

$$\| |W_s S| + |WT| \|_\infty < \sqrt{2} \left\| \begin{bmatrix} WT \\ W_s S \end{bmatrix} \right\|_\infty \quad (\text{A.44})$$

Therefore, the approximate solution is different from the real solution by at most a factor of $\sqrt{2}$. Doyle et al. [35], presented a solution to the general mixed-sensitivity problem in 1989 using state-space representations. The μ -synthesis Toolbox of Matlab incorporates this solution [10]. Here we present only the system representation used to solve the problem and leave the technical details to [35, 41].

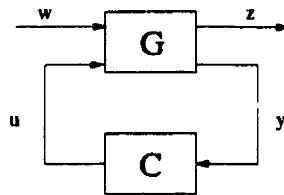


FIGURE A.8. The basic design configuration for the generalized plant G

Figure A.8 illustrates the basic design configuration for the system for mixed-sensitivity problem, in which $\mathbf{w}$ is the vector of exogenous inputs to the generalized plant $\mathbf{G}$, $\mathbf{z}$ is the vector of regulated outputs, $\mathbf{y}$ is the vector of measured outputs, and $\mathbf{u}$ is the vector of control inputs to the plant. The generalized plant G absorbs the plant P and the weighting functions. The design objective is to keep the $\mathcal{H}_\infty$ norm of the transmission $\mathbf{T}_{wz}$, the transfer matrix from vector input $\mathbf{w}$ to the vector output $\mathbf{z}$, small. It is illustrated in § 3 of Chapter 6 how the plant P and weighting functions W_s , W_u , and W can be arranged such that the transfer matrix from the only input y_d to vector outputs $\mathbf{z}$ is given by

$$\mathbf{T}_{wz} = T_{y_d z} = \begin{bmatrix} WT \\ W_s S \\ W_u U \end{bmatrix} \quad (\text{A.45})$$

where U is the transfer function from the desired output y_d to the control effort u (Refer to Equation A.37, $U = P^{-1}T$). Solving the mixed-sensitivity problem for this case guarantees $\|\mathbf{T}_{wz}\|_\infty < 1$, by which the weighted sensitivity and weighted complementary sensitivity is

minimized while constraining the actuator effort. In this way robust stability is preserved, and the closed-loop performance is guaranteed within the physical saturation limits of the actuator.

Current solution algorithms [35, 41], are based on state-space representation. The general plant G can be represented as:

$$\begin{aligned}\dot{\mathbf{x}} &= A\mathbf{x} + B_1\mathbf{w} + B_2\mathbf{u} \\ \mathbf{z} &= C_1\mathbf{x} + D_{11}\mathbf{w} + D_{12}\mathbf{u} \\ \mathbf{y} &= C_2\mathbf{x} + D_{21}\mathbf{w} + D_{22}\mathbf{u}\end{aligned}\tag{A.46}$$

This representation combines the system nominal model and the weighting functions. In numerical software generalized plant is represented by the following matrix:

$$G(s) = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array} \right] = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right]\tag{A.47}$$

Doyle et al. [35], give a solution for $\|\mathbf{T}_{wz}\|_\infty < \gamma$ under the following assumptions:

(A1) (A, B_2, C_2) is stabilizable and detectable.

This is required for the existence of a stabilizing C .

(A2) $D_{12}^T D_{12}$ and $D_{21} D_{21}^T$ are nonsingular.

This is sufficient to ensure that the controllers are proper.

(A3) (A, B_1, C_1) is stabilizable and detectable.

Assumption (A3) disallows uncontrollable and unobservable modes of (A, B_1, C_1) on the imaginary axis. Assumption (A3) is slightly modified in [41] into

$$(A3) \left[\begin{array}{cc} A - j\omega I & B_2 \\ C_1 & D_{12} \end{array} \right] \text{ and } \left[\begin{array}{cc} A^T - j\omega I & C_2^T \\ B_1^T & D_{21}^T \end{array} \right] \text{ have full column rank } \forall \omega$$

where in this assumption open RHP uncontrollable or unobservable modes are allowed. The solution to mixed-sensitivity problem involves the solution of two algebraic Riccati equations, each with the same order as the generalized plant G , and further gives feasible controllers also with this order. The implementation of this solution⁴ in μ -synthesis toolbox of Matlab [10], provides iteration in γ using the bisection method, in an effort to approach the optimal $\mathcal{H}_\infty$ solution. The search for minimum γ is carried out by decreasing γ until a solution fails to exist.

⁴Function *hinfsyn* in μ -synthesis Toolbox of Matlab

APPENDIX B

Software Code Listings

1. Open-Loop Time Response

1.1. The Host Portion

```
/*-----
*   FILE: load_constant_time.c
*
*   Calling Sequence: Main program.
*
*   Function: Interactive loader for sampling at constant time intervals.
*
*   Revision Information:
*       Date      By   Changes Made....
*       22/06/92   ptd  Modified loadadlights.c
*       24/02/94   ptd  Conversions of sample data from Ti_float format.
*       20/10/95   hdt  Updating the reference signals to incorporate
*                       duty-cycled functions.
*-----*/
#include <sys/errno.h>
#include <stdio.h>                /* contains declaration of scanf() */
#include <stdlib.h>               /* declaration of atof(); extern double atof() */
#include <malloc.h>
#include "skyc30v.h"
#include "stypes.h"
#include "ti2ie3cnv.h"

#define NODE1      1                /* C30 node */
#define NODE2      2                /* C30 node */
#define CARD       0                /* the Challenger card number */
#define GLOBAL     0                /* for GRAM */

#define NODE1FILE   "constant_time.out" /* C30 program */
#define NODE2FILE   "c30loop.out"      /* C30 program */
#define GMEM_BASE   0xc00000          /* Global memory */
#define GMEM_OFFSET_PP 0xc0          /* C30 Parameter Passing space*/
#define GMEM_OFFSET_SD 0x3000        /* C30 Sample Data space */
#define FIELD_QTY   4                /* Number of fields per sample*/
```

```

extern TiFloat ie3_to_ti();
extern float ti_to_ie3();
SKYC30_device *sky_open_card();

long *samples;
/* ----- */
main()
{
    unsigned long i,delay, numsamples, wavetype;
    BIT_32 *addr;
    int dummy, status, channel, numchannels = FIELD_QTY;
    char fname1[80];
    char inputvar1[80], inputvar2[80];
    FILE *out_file1;
    float Nbias, Nref, sample_period, dcd_r_dec, freq, duty;
    Bool Samples_Ready = False;
    TiFloat tmpf_Ti, Nbias_Ti, Nref_Ti, sample_period_Ti, freq_Ti, duty_Ti;
    char *src, *dst;
    short dcd_rvalue;
    SKYC30_device *device; /* pointer to the hardware */

    /* Acquire the card and make sure that the card is not doing anything. */
    out_file1 = fopen("samples.dat","w");
    device = sky_open_card(0); /* assign the card */
    if ( device == NULL )
    { perror("sky_open_card: ");
      exit(-1);
    }
    sky_nodes_off(0); /* turn it off */

    /* Load the C30 test program into node 1. */
    /* printf(" Filename to load in node 1 : ");
    scanf("%s",fname1); */
    /* status = sky_ld_file (0, NODE1, fname1); */
    status = sky_ld_file (0, NODE1, NODE1FILE);
    if (status != 0)
    {
        printf ("Node 1, sky_ld_file() for file '%s' failed; status = %d\n",
                fname1, status);
        exit (-1);
    }

    /* Load the C30 test program into node 2. */
    status = sky_ld_file (0, NODE2, NODE2FILE);
    if (status != 0)
    {
        printf ("Node 2, sky_ld_file() for file '%s' failed; status = %d\n",
                NODE2FILE, status);
        exit (-1);
    }

    /* Parameter passing via C30 global memory. Floating point conversion of
    the parameter passed by byte-wise copy of the bit pattern */
    addr = device->mem + GMEM_OFFSET_PP;
    printf(" Enter speed/current reference bias (+/- %%): ");
    scanf("%s",inputvar1);

```



```

Nbias = atof(inputvar1);
tmpf_Ti = ie3_to_ti( &Nbias );
src = (char *) &tmpf_Ti;
dst = (char *) &Nbias_Ti;
*dst++ = *src++; *dst++ = *src++; *dst++ = *src++; *dst++ = *src++;
status = sky_wr_mem (CARD, GLOBAL, addr, 1, &Nbias_Ti);
if (status != 0)
{ printf ("sky_wr_mem() failed; status = %d\n", status);
  exit (-1);
}

/* Parameter passing via C30 global memory. */
addr = device->mem + GMEM_OFFSET_PP + 1;
status = sky_wr_mem (CARD, GLOBAL, addr, 1, &Samples_Ready);
if (status != 0)
{ printf ("sky_wr_mem() failed; status = %d\n", status);
  exit (-1);
}

/* Parameter passing via C30 global memory. */
addr = device->mem + GMEM_OFFSET_PP + 2;
printf("      Enter sample period Ts (s):  ");
scanf("%s",inputvar2);
sample_period = atof(inputvar2);
tmpf_Ti = ie3_to_ti( &sample_period );
src = (char *) &tmpf_Ti;
dst = (char *) &sample_period_Ti;
*dst++ = *src++; *dst++ = *src++; *dst++ = *src++; *dst++ = *src++;
status = sky_wr_mem (CARD, GLOBAL, addr, 1, &sample_period_Ti);
if (status != 0)
{printf ("sky_wr_mem() failed; status = %d\n", status);
  exit (-1);
}

/* Parameter passing via C30 global memory. */
addr = device->mem + GMEM_OFFSET_PP + 3;
printf("      Enter the number of samples:  ");
scanf("%s",inputvar1);
numsamples = atof(inputvar1);
samples = (long *) malloc( (numsamples*FIELD_QTY) * sizeof(long) );
status = sky_wr_mem (CARD, GLOBAL, addr, 1, &numsamples);
if (status != 0)
{printf ("sky_wr_mem() failed; status = %d\n", status);
  exit (-1);
}

/* Parameter passing via C30 global memory. Floating point conversion of
the parameter passed by byte-wise copy of the bit pattern */
addr = device->mem + GMEM_OFFSET_PP + 4;
printf("      Enter speed/current reference amplitude (%):  ");
scanf("%s",inputvar1);
Nref = atof(inputvar1);
tmpf_Ti = ie3_to_ti( &Nref );
src = (char *) &tmpf_Ti;
dst = (char *) &Nref_Ti;
*dst++ = *src++; *dst++ = *src++; *dst++ = *src++; *dst++ = *src++;

```

```

    status = sky_wr_mem (CARD, GLOBAL, addr, 1, &Nref_Ti);
    if (status != 0)
    { printf ("sky_wr_mem() failed; status = %d\n", status);
      exit (-1);
    }

    /* Parameter passing via C30 global memory. */
    addr = device->mem + GMEM_OFFSET_PP + 5;
    printf("      Enter the waveform : \n    1 - sine\n    2 - square\n    3 -
triangle\n    4 - inverted sine \n    5 - half sine\n    6 - inverted half
sine\n    7 - composite\n    8 - asy_sine\n    9 - Duty-cycled sine\n
10- Duty-cycled square\n    11- Duty-cycled triangle\n");
    scanf("%s",inputvar1);
    wavetype = atof(inputvar1);
    status = sky_wr_mem (CARD, GLOBAL, addr, 1, &wavetype);
    if (status != 0)
    {printf ("sky_wr_mem() failed; status = %d\n", status);
      exit (-1);
    }

    /* Parameter passing via C30 global memory. */
    addr = device->mem + GMEM_OFFSET_PP + 6;
    printf("      Enter the waveform frequency (Hz):  ");
    scanf("%s",inputvar1);
    freq = atof(inputvar1);
    tmpf_Ti = ie3_to_ti( &freq );
    src = (char *) &tmpf_Ti;
    dst = (char *) &freq_Ti;
    *dst++ = *src++; *dst++ = *src++; *dst++ = *src++; *dst++ = *src++;
    status = sky_wr_mem (CARD, GLOBAL, addr, 1, &freq_Ti);
    if (status != 0)
    {
      printf ("sky_wr_mem() failed; status = %d\n", status);
      exit (-1);
    }

    /* Parameter passing via C30 global memory. */
    if ( wavetype > 8)
    {
      addr = device->mem + GMEM_OFFSET_PP + 7;
      printf("      Enter the duty cycle ratio:  ");
      scanf("%s",inputvar1);
      duty = atof(inputvar1);
      tmpf_Ti = ie3_to_ti( &duty );
      src = (char *) &tmpf_Ti;
      dst = (char *) &duty_Ti;
      *dst++ = *src++; *dst++ = *src++; *dst++ = *src++; *dst++ = *src++;
      status = sky_wr_mem (CARD, GLOBAL, addr, 1, &duty_Ti);
      if (status != 0)
      {
        printf ("sky_wr_mem() failed; status = %d\n", status);
        exit (-1);
      }
    }

    /* Parameter passing via C30 global memory. */
    if ( wavetype == 8)

```

```

{
    addr = device->mem + GMEM_OFFSET_PP + 7;
    printf("      Enter the Asymetric ratio: ");
    scanf("%s",inputvar1);
    duty = atof(inputvar1);
    tmpf_Ti = ie3_to_ti( &duty );
        src = (char *) &tmpf_Ti;
        dst = (char *) &duty_Ti;
        *dst++ = *src++; *dst++ = *src++; *dst++ = *src++; *dst++ = *src++;
    status = sky_wr_mem (CARD, GLOBAL, addr, 1, &duty_Ti);
    if (status != 0)
    {
        printf ("sky_wr_mem() failed; status = %d\n", status);
        exit (-1);
    }
}

/* Run the C30 programs. */
sky_reset_go(0);
addr = device->mem + GMEM_OFFSET_PP + 0x1;
while( Samples_Ready == False ){          /* Samples ready??? */
    printf("      Waiting for Samples_Ready flag...\n");
    for (delay=0; delay< (numsamples* 8 * sample_period/0.0001)
        ; delay++);
    status = sky_rd_mem(CARD,GLOBAL,addr,1,&Samples_Ready);
    if (status != 0)
    {
        printf("sky_rd_mem failed \n");
        exit (-1);
    }
}
printf("      Sample data upload to host in progress...\n");

/* Fetch the C30 sample data and write to output */
addr = device->mem + GMEM_OFFSET_SD;
status = sky_rd_mem(CARD,GLOBAL,addr,(numsamples*FIELD_QTY),samples);
if (status != 0)
{
    printf("sky_rd_mem failed \n");
    exit (-1);
}

/* Fetch the C30 sample data and write to output */
for (i=0; i < numsamples; i++) {
    for (channel = 0; channel < numchannels; channel++){
        fprintf(out_file1,"%12.8f\n", ti_to_ie3( samples )) ;
        samples++;
    }
}
fclose(out_file1);
}
/* ----- End of Module ----- */

```

1.2. The C30 Portion

```

/*-----
 * FILE: constant_time.c
 *
 * Calling Sequence:
 *
 * Function: C30 node "C" program to respond to C30V timer-generated
 *           interrupts using timer 0 and sample a2d inputs and the pulse count
 *           from the quadrature decoder.
 *
 * Revision Information:
 *   Date      By      Changes Made....
 *
 *   26/02/92  ptd     Cleaned up and documented.
 *   11/03/92  ptd     Expanded from 2 channels to four. WORKS, also.
 *   24/02/94  ptd     DT140X analog input boards; float functionalities.
 *   20/10/95  hdt     Cleaned up & duty-cycled reference signals included
 *
 * ***** */

#include "DT140X.h"          /* Analog-to-digital converter definitions */
#include <math.h>             /* Sine-wave generator functions */
#include "node_map.h"        /* C30 node address definitions */
#include "stypes.h"          /* Boolean type definition */
#include "servo_refgen.h"     /* Servo reference generation functions */
#include "servo_conv.h"      /* Servo variable value definitions */

#define FIELD_QTY      4          /* Number of fields per sample */
#define NUM_A2D_CHAN    DT1401_A2D_CHANNELS

short *mbox = (short *) 0xe0;          /* Mail boxes */
float *parm_ptr = (float *) 0xc000c0; /* Parameter passing mailboxes */
float *sample_ptr = (float *) 0xc03000; /* Sample data storage */
float Namplitude = 0.0, Vamplitude = 0.0;
float Nbias = 0.0, Vbias = 0.0;
float theta = 0, freq = 0 ; /* Angle and angular frequency of function */
float rise = 0.0;
float fall = 0.0;
float duty = 1.0;          * duty cycle ratio for some reference inputs */

void c_int10();

DT140XChan
    str_ch = DT140X_CH3,
    end_ch = DT140X_CH6;
int numchannels = FIELD_QTY;          /* number of a2d channels */
int wavetype = 0;                     /* no wave generation by default */
int samplecount = 0;                  /* sample count */
float sample_period = 0.001;          /* Sample period in decimal */

/*----- Timer 1 ETINT1 handler -----*/
void c_int10()
{

```

```

float a2dsamples[NUM_A2D_CHAN];      /* current set of resolved values */
int channel, Nref;
float ref_fl;                        /* reference to MAX-100 servo */
DT1401Read( str_ch, end_ch, a2dsamples);
ref_fl = reference_generation(wavetype); /* floatV for MAX-100 servo */
write_dac0( ref_fl );

/* writing the wanted variables into Gram for further use */
copy(&ref_fl, sample_ptr++,1,0x0000, NRAM_2_GRAM);
for (channel = str_ch+1; channel < (str_ch+numchannels); channel++)
    copy(&a2dsamples[channel], sample_ptr++,1,0x0000, NRAM_2_GRAM);

++samplecount;
enable_intr();
} /* c_int10() */

/* ***** */
void main()
{
    Bool Samples_Ready = True;
    int channel, pause, numsamples;

    copy(parm_ptr,  &Nbias, 1, 0x0000, 21);          /* GRAM->NRAM */
    copy(parm_ptr+2, &sample_period, 1, 0x0000, 21); /* GRAM->NRAM */
    copy(parm_ptr+3, &numsamples, 1, 0x0000, 21);    /* GRAM->NRAM */
    copy(parm_ptr+4, &Namplitude, 1, 0x0000, 21);    /* GRAM->NRAM */
    copy(parm_ptr+5, &wavetype, 1, 0x0000, 21);      /* GRAM->NRAM */
    copy(parm_ptr+6, &freq, 1, 0x0000, 21);          /* GRAM->NRAM */
    copy(parm_ptr+7, &duty, 1, 0x0000, 21);          /* GRAM->NRAM */

    Vbias = PC2V_linear_conv( Nbias );
    Vamplitude = PC2V_linear_conv( Namplitude );
    setup_dac_csr();
    write_dac0( Vbias );

    for (pause=0; pause < 0xffff; pause++);

    /* Initialize a2d converter; sample rate & mapping */
    DT140XInit(str_ch, end_ch);

    /* Main interrupter routine */
    t1_schedule_intr(c_int10,sample_period);
    enable_intr();
    t1_start();

    while( samplecount < numsamples ){}
    disable_intr();

    copy(&Samples_Ready,(parm_ptr +1),1,0x0000, 16); /* NRAM->GRAM */
    write_dac0(0.0);
    c30_led_sflash();
}

/* ----- End of Module ----- */

```

2. Torque Ripple Estimation by Kalman Filter

```

/*-----
* FILE: free_kalman.c
*
* Calling Sequence:
*
* Function: 1- C30 node "C" program to respond to C30V timer-generated
*           interrupts using timer 0 and sample a2d inputs and the pulse count
*           from the quadrature decoder.
*           2- encoder pulse count acquisition.
*           3- Torque ripples filtered by Kalman Filter
*
* Revision Information:
*   Date      By    Changes Made....
*
*   26/02/92  ptd   Cleaned up and documented.
*   11/03/92  ptd   Expanded from 2 channels to four.
*   24/02/94  ptd   DT140X analog input boards; float functionalities.
*   01/11/95  hdt   Butterworth filter for torque
*   18/04/96  hdt   Differentiator Filter for acceleration
*   18/04/96  hdt   Kalman filter estimation for torque ripples
*
***** */

#include "DT140X.h"          /* Analog-to-digital converter definitions */
#include <math.h>              /* Sine-wave generator functions */
#include "node_map.h"         /* C30 node address definitions */
#include "stypes.h"           /* Boolean type definition */
#include "servo_refgen.h"     /* Servo reference generation functions */
#include "servo_conv.h"       /* Servo variable value definitions */

#define NUM_CHANNELS 4
#define NUM_MISC_FIELDS 1
#define FIELD_QTY (NUM_CHANNELS+NUM_MISC_FIELDS)
/* Number of fields per sample */
#define NUM_A2D_CHAN DT1401_A2D_CHANNELS
#define SIGN(X) ((X) < 0 ? -1 : 1 )

int *time_sample_ptr = (int *) 0xc23000; /* Sample data storage */
short *mbox = (short *) 0xe0; /* Mail boxes */
float *parm_ptr = (float *) 0xc000c0; /* Parameter passing mailboxes */
float *sample_ptr = (float *) 0xc03000; /* Sample data storage */
float Namplitude = 0.0, Vamplitude = 0.0;
float Nbias = 0.0, Vbias = 0.0;
float theta = 0, freq = 0 ; /* Angle and angular frequency of function */
float rise = 0.0;
float fall = 0.0;
float duty = 1.0; /* duty cycle ratio for some reference inputs */

void c_int10();
float Diff_filter();
float Kalman();

```

```

/* Constant needed for Subroutines */
int kal_count = 1;
float v_1=0.0, a_1=0.0;
float volt2Nm = 6.6, volt2radps = 56.2565;
float J_load = 1.725e-2, Gain = 100;
float T_ripple = 0.0;

DT140XChan
    str_ch = DT140X_CH3,
    end_ch = DT140X_CH6;
int numchannels = NUM_CHANNELS;           /* number of a2d channels */
int wavetype = 0;                         /* no wave generation by default */
int samplecount = 0;                      /* sample count */
float sample_period = 0.001;              /* Sample period in decimal */
int time_samplecount = 0;

/* ***** */
void main()
{
    Bool Samples_Ready = True;
    int channel, pause, numsamples;
    float tperiod = 514.0;                /* max. allowable Timer period in */
                                          /* seconds */
    copy(parm_ptr, &Nbias, 1, 0x0000, 21); /* GRAM->NRAM */
    copy(parm_ptr+2, &sample_period, 1, 0x0000, 21); /* GRAM->NRAM */
    copy(parm_ptr+3, &numsamples, 1, 0x0000, 21); /* GRAM->NRAM */
    copy(parm_ptr+4, &Namplitude, 1, 0x0000, 21); /* GRAM->NRAM */
    copy(parm_ptr+5, &wavetype, 1, 0x0000, 21); /* GRAM->NRAM */
    copy(parm_ptr+6, &freq, 1, 0x0000, 21); /* GRAM->NRAM */
    copy(parm_ptr+7, &duty, 1, 0x0000, 21); /* GRAM->NRAM */

    /* Initialize the hardware "my addition" */

    xvme085_init(); /* Prototype board encoder interface initialized */
    init_c30v_bus(); /* C30 node and buses initialized */
    c30_led_on(); /* Port configuration reset, mode 0 */
    t0_schedule_intr(c_int10, sample_period); /* CPU interrupt level 0x9 */
    enable_ie_int3(); /* CPU external interrupt #3 enabled */

    Vbias = PC2V_linear_conv( Nbias );
    Vamplitude = PC2V_linear_conv( Namplitude );
    setup_dac_csr();
    write_dac0( Vbias );

    /* Allow for mech. time constant */
    for (pause=0; pause < 0xffff; pause++);

    /* setup a2d converter; sample rate & mapping */
    DT140XInit(str_ch, end_ch);
    t1_schedule_intr(c_int10, sample_period);
    enable_intr();
    t1_start();

    while( samplecount < numsamples ){
        disable_intr();
        copy(&time_samplecount, (parm_ptr + 10), 1, 0x0000, NRAM_2_GRAM);
    }
}

```

```

    copy(&Samples_Ready,(parm_ptr +1),1,0x0000, 16); /* NRAM->GRAM */
    write_dac0(0.0);
    c30_led_sflash();
}
/* ***** Misc Sub_Routines ***** */

void c_int10()
{
    int *t1_counter_reg = (int *) T1_CNTR_REG;
    /* timemark = blank dummy */
    int channel, timecnt, timemark = 0xffffffff;
    /* current set of resolved values */
    float a2dsamples[NUM_A2D_CHAN];
    float Tacho, MotorCurrOut, Torque, Torque_f, ref_fl;
    float vel, acc, T_meas, T_mod, Delta_T, out;
    short pcsample;
    short EncoderPulses;
    int Nref;

    t1_stop();
    timecnt = *t1_counter_reg;
    t1_start();
    copy( PB_LAT_RD_ADDR,&pcsample,1,0xca00,VME_2_NRAM);

    DT1401Read( str_ch, end_ch, a2dsamples);

    Tacho      = a2dsamples[DT140X_CH4];
    MotorCurrOut = a2dsamples[DT140X_CH5];
    Torque      = a2dsamples[DT140X_CH6];

    vel      = Tacho * volt2radps; /* Calibration volt --> rad/sec */
    acc      = Diff_filter( vel); /* Differentiation of velocity */
    T_meas   = Torque * volt2Nm; /* Calibration volt --> Nm */
    T_mod    = J_load * acc / Gain; /* Evaluatind Modelled Torque */
    Delta_T  = T_meas - T_mod; /* Torque Error */
    Torque_f  = T_meas - T_ripple; /* Final Filtered Torque */
    T_ripple  = Kalman (Delta_T, vel, sample_period);
                                /* Finding torque ripple for next
                                iteration: Prediction-Type KF */
    ref_fl=reference_generation(wavetype);/* floatV for MAX-100 servo */
    write_dac0( ref_fl );

    /* writing the wanted variables into Gram for further use */
    copy(&T_meas, sample_ptr++,1,0x0000, NRAM_2_GRAM);
    copy(&vel, sample_ptr++,1,0x0000, NRAM_2_GRAM);
    copy(&MotorCurrOut, sample_ptr++,1,0x0000, NRAM_2_GRAM);
    copy(&Torque_f, sample_ptr++,1,0x0000, NRAM_2_GRAM);

    /****** */

    /* Store the interpulse time data */
    copy(&timecnt,time_sample_ptr++,1,0x0000, NRAM_2_GRAM); /* NRAM->GRAM */
    /* Process and store the pulse count data */
    pcsample &= 0x0000ffff; /* Avoid hi/lo duplication of short */
    copy(&pcsample, sample_ptr++,1,0x0000, NRAM_2_GRAM); /* NRAM->GRAM */
    EncoderPulses = pcsample;

```


2. TORQUE RIPPLE ESTIMATION BY KALMAN FILTER

```

copy(&timemark,time_sample_ptr++,1,0x0000,NRAM_2_GRAM);
time_samplecount = time_samplecount + 3;
++samplecount;
enable_intr();
} /* c_int10() */

/***** Differentiator Filter *****/
*
* Discretized Differentiator Filter  $H(s) = s / (0.005 s + 1)$ 
* Discretized by tustin method for  $T_s = 1$  ms
*
* Function Calls : a = Diff_filter (v)
*
*   input :      v = velocity      (rad/s)
*   output:      a = acceleration (rad/s/s)
*****/

float Diff_filter (v)
float v;
{
    float nd[2]={
        181.8181818182,
        -181.8181818182
    };
    float dd[2]={
        1.00000000000000,
        -0.818181818182
    };
    float a;

    a=nd[0]*v + nd[1]*v_1 - dd[1]*a_1;
    v_1 = v; a_1 = a;

    return(a);
}

/* *****/
*
* Kalman Filter estimation for HD Torque ripples
* function calls
*
*   x = KALMAN ( y , v , T)
*
*   y : T_meas - T_mod    ( your estimate of the torque error)
*   v : velocity (rad/sec)
*   T : Sampling Period (sec)
*   x : The estimate of the torque ripple
*
* Algorithm:
*
* A second order harmonic oscillator, with two harmonics of
* oscillation equal to twice and for times velocity is the model
* of the torque ripples. Time varying Kalman Filter is considered
* to estimate the torque ripple.
*
* Last Update : April 18, 1996
* By: Hamid D. Taghirad

```

```

*
*****/

float Kalman( Delta_T, velocity, sample_period)

float Delta_T;
float velocity;
float sample_period;

{
    int i, j, k;
    float om1, om2, K[4], PC[4], pPC[4], CPC;
    float phi [4][4], Pp[4][4], KC[4][4], p_KC[4][4], gb[4][4];
    float x1[4], Torque_ripple;
    float x[4], C[4], R, P[4][4], Q[4][4];

    /* Matrix Initialization */

    if (kal_count == 1)    /* only true for first iteration */
    {
        R = 1.0;
        for(i=0; i<4; ++i)
        {
            x[i] = 0.0;
            C[i] = 0.0;
            for(j=0; j< 4; ++j)
            {
                if ( i == j )
                {
                    P[i][j] = 1.0;
                    Q[i][j] = 1000.0;
                }
                else
                {
                    P[i][j] = 0.0;
                    Q[i][j] = 0.0;
                }
            }
        }
        x[0] = x [2] = Delta_T;
        C[0] = C [2] = 1.0;

        ++kal_count;
    }
    /* Two harmonics of the Torque ripples * Sample Period */
    om1 = 2.0 * velocity * sample_period;
    om2 = 2.0 * om1;

    /* System Model of Harmonic Oscillator 4th order */
    for(i=0; i< 4; ++i)
    {
        for(j=0; j<4; ++j)
            phi[i][j]=0.0;
    }
    phi[0][0] = cos(om1);
    phi[0][1] = sin(om1);

```

```

phi[1][0] = -sin(om1);
phi[1][1] = cos(om1);
phi[2][2] = cos(om2);
phi[2][3] = sin(om2);
phi[3][2] = -sin(om2);
phi[3][3] = cos(om2);

/* ***** 1 ***** */
/* Gain update K */

/* Matrix Multiplication PC(4x1) = P(4x4) * C'(4x1) */
for ( i=0; i < 4; ++i)
{
    PC[i] = 0.0;
    for(j=0; j < 4; ++j)
        PC[i] += P[i][j] * C[j];
}

/* Matrix Multiplication pPC(4x1) = phi(4x4) * PC(4x1) */
for ( i=0; i < 4; ++i)
{
    pPC[i] = 0.0;
    for(j=0; j < 4; ++j)
        pPC[i] += phi[i][j] * PC[j];
}

/* Matrix Multiplication CPC(1x1) = C(1x4) * PC(4x1) */
CPC = 0.0;
for(j=0; j < 4; ++j)
    CPC += C[j] * PC[j];

/* Final Gain Calculation K(4x1) = phi P C' / ( R + C P C' )
 * Note that (R + C P C') is scalar --> NO Matrix inversion
 */

for ( i=0; i < 4; ++i)
    K[i] = pPC[i] / ( R + CPC);

/* ***** 2 ***** */
/* P Update */

/* Matrix Multiplication Pp(4x4) = P(4x4) * phi'(4x4) */
for ( i=0; i < 4; ++i)
{
    for(j=0; j < 4; ++j)
    {
        Pp[i][j] = 0.0;
        for(k = 0; k < 4 ; ++k)
            Pp[i][j] += P[i][k] * phi[j][k];
    }
}

/* Matrix Multiplication KC(4x4) = K(4x1) * C(1x4) */
for ( i=0; i < 4; ++i)
{

```

```

    for(j=0; j < 4; ++j)
        KC[i][j] = K[i] * C[j];
}

/* Matrix Addition p_KC(4x4) = ( phi(4x4) - KC(4x4)) */
for ( i=0; i < 4; ++i)
{
    for(j=0; j < 4; ++j)
        p_KC[i][j] = phi[i][j] - KC[i][j];
}

/* Matrix Multiplication gb(4x4) =(phi - K C)P phi'=p_KC(4x4)*Pp(4x4) */
for ( i=0; i < 4; ++i)
{
    for(j=0; j < 4; ++j)
    {
        gb[i][j] = 0.0;
        for(k = 0; k < 4 ; ++k)
            gb[i][j] += p_KC[i][k] * Pp[k][j];
    }
}

/* Final formula for P(4x4) = Q + (phi - K C) P phi' = Q + gb */
for ( i=0; i < 4; ++i)
{
    for(j=0; j < 4; ++j)
        P[i][j] = Q[i][j] + gb[i][j];
}

/* ***** 3 ***** */
/* x Update, and output generation */

/* Matrix Multiplication x1(4x1) = ( phi -KC ) * x = p_KC(4x4) * x(4x1) */
for ( i=0; i < 4; ++i)
{
    x1[i] = 0.0;
    for(j=0; j < 4; ++j)
        x1[i] += p_KC[i][j] * x[j];
}
/* Matrix Multiplication x(4x1) = (phi -KC) x+K y=K(4x1)*y(1x1)+x1(4x1) */
for ( i=0; i < 4; ++i)
    x[i] = K[i] * Delta_T + x1[i];

/* Output generation y = C x = x(1) + x(3) */
Torque_ripple = x[0] + x[2] ;

return(Torque_ripple);
}

/* -- end of file free_kalman.c --*/

```

3. Torque Ripple and Misalignment Torque Estimation

```

/* ***** Kalman *****
*
*      Kalman Filter estimation for HD Torque ripples
*      function calls
*
*      x = KALMAN ( y , v , T)
*
*      y : T_meas - T_mod    ( your estimate of the torque error)
*      v : velocity (rad/sec)
*      T : Sampling Period (sec)
*      x : The estimate of the torque ripple
*
*      Algorithm:
*
*      A second order harmonic oscillator, with two harmonics of
*      oscillation equal to twice and for times velocity is the model
*      of the torque ripples. Time varying Kalman Filter is considered
*      to estimate the torque ripple.
*
*
*      Last Update : April 18, 1996
*      By: Hamid D. Taghirad
*
*****/

float Kalman( Delta_T, velocity, sample_period)

float Delta_T;
float velocity;
float sample_period;

{
    int i, j, k;
    float om1, om2, K[4], PC[4], pPC[4], CPC;
    float phi [4][4], Pp[4][4], KC[4][4], p_KC[4][4], gb[4][4];
    float x1[4], Torque_ripple;
    float x[4], C[4], R, P[4][4], Q[4][4];

    /* Matrix Initialization */

    if (kal_count == 1)    /* only true for first iteration */
    {
        R = 1.0;
        for(i=0; i<4; ++i)
        {
            x[i] = 0.0;
            C[i] = 0.0;
            for(j=0; j< 4; ++j)
            {
                if ( i == j )
                {
                    P[i][j] = 1.0;

```

```

        Q[i][j] = 1000.0;
    }
    else
    {
        P[i][j] = 0.0;
        Q[i][j] = 0.0;
    }
}
}
x[0] = x [2] = Delta_T;
C[0] = C [2] = 1.0;
Q[2][2] = Q[3][3] = 5e-2;

++kal_count;
}

/* Two harmonics of the Torque ripples * Sample Period */
om1 = 2.0 * velocity * sample_period;
om2 = velocity * sample_period / 100.0; /* 100 == HD Gear ratio */

/* System Model of Harmonic Oscillator 4th order */
for(i=0; i< 4; ++i)
{
    for(j=0; j<4; ++j)
        phi[i][j]=0.0;
}

phi[0][0] = cos(om1);
phi[0][1] = sin(om1);
phi[1][0] =-sin(om1);
phi[1][1] = cos(om1);

phi[2][2] = cos(om2);
phi[2][3] = sin(om2);
phi[3][2] =-sin(om2);
phi[3][3] = cos(om2);

/* ***** 1 ***** */
/* Gain update K */

/* Matrix Multiplication PC(4x1) = P(4x4) * C'(4x1) */
for ( i=0; i < 4; ++i)
{
    PC[i] = 0.0;
    for(j=0; j < 4; ++j)
        PC[i] += P[i][j] * C[j];
}

/* Matrix Multiplication pPC(4x1) = phi(4x4) * PC(4x1) */
for ( i=0; i < 4; ++i)
{
    pPC[i] = 0.0;
    for(j=0; j < 4; ++j)
        pPC[i] += phi[i][j] * PC[j];
}

```

```

}

/* Matrix Multiplication CPC(1x1) = C(1x4) * PC(4x1) */
CPC = 0.0;
for(j=0; j < 4; ++j)
    CPC += C[j] * PC[j];

/* Final Gain Calculation K(4x1) = phi P C' / ( R + C P C' )
 * Note that ( R + C P C' ) is scalar --> NO Matrix inversion
 */

for ( i=0; i < 4; ++i)
    K[i] = pPC[i] / ( R + CPC);

/* ***** 2 ***** */
/* P Update */

/* Matrix Multiplication Pp(4x4) = P(4x4) * phi'(4x4) */
for ( i=0; i < 4; ++i)
{
    for(j=0; j < 4; ++j)
    {
        Pp[i][j] = 0.0;
        for(k = 0; k < 4 ; ++k)
            Pp[i][j] += P[i][k] * phi[j][k];
    }
}

/* Matrix Multiplication KC(4x4) = K(4x1) *C(1x4) */
for ( i=0; i < 4; ++i)
{
    for(j=0; j < 4; ++j)
        KC[i][j] = K[i] * C [j];
}

/* Matrix Addition p_KC(4x4) = ( phi(4x4) - KC(4x4)) */
for ( i=0; i < 4; ++i)
{
    for(j=0; j < 4; ++j)
        p_KC[i][j] = phi[i][j] - KC[i][j];
}

/* Matrix Multiplication gb(4x4) =(phi - K C)P phi' =p_KC(4x4)*Pp(4x4) */
for ( i=0; i < 4; ++i)
{
    for(j=0; j < 4; ++j)
    {
        gb[i][j] = 0.0;
        for(k = 0; k < 4 ; ++k)
            gb[i][j] += p_KC[i][k] * Pp[k][j];
    }
}

/* Final formula for P(4x4) = Q + (phi - K C) P phi' = Q + gb */
for ( i=0; i < 4; ++i)

```

3. TORQUE RIPPLE AND MISALIGNMENT TORQUE ESTIMATION

```

{
    for(j=0; j < 4; ++j)
        P[i][j] = Q[i][j] + gb[i][j];
}

/* ***** 3 ***** */
/* x Update, and output generation */

/* Matrix Multiplication x1(4x1) = ( phi -KC ) * x = p_KC(4x4) * x(4x1) */
for ( i=0; i < 4; ++i)
{
    x1[i] = 0.0;
    for(j=0; j < 4; ++j)
        x1[i] += p_KC[i][j] * x[j];
}
/* Matrix Multiplication x(4x1) = (phi -KC) x+K y=K(4x1)*y(1x1)+x1(4x1) */
for ( i=0; i < 4; ++i)
    x[i] = K[i] * Delta_T + x1[i];

/* Output generation y = C x = x(1) + x(3) */
Torque_ripple = x[0] + x[2] ;

return(Torque_ripple);
}
/* ----- End of Module ----- */

```


4. Closed-Loop Time Response

4.1. Kalman Filtered Torque, $\mathcal{H}_\infty$ Controller and Friction Compensator

```

/*-----
* FILE: fric_kfdr_hinf.c
*
* Calling Sequence:
*
* Function: 1- C30 node "C" program to respond to C30V timer-generated
*           interrupts using timer 0 and sample a2d inputs and the pulse count
*           from the quadrature decoder.
*           2- encoder pulse count acquisition.
*           3- Applies friction compensation feedback
*           4- Kalman filter estimation of the torque measurements
*           5- H-infinity controller implementation
*
* Revision Information:
*   Date      By      Changes Made....
*
*   26/02/92  ptd     Cleaned up and documented.
*   11/03/92  ptd     Expanded from 2 channels to four.
*   24/02/94  ptd     DT140X analog input boards; float functionalities
*   12/10/95  hdt     Friction compensation loop added
*   29/10/96  hdt     Boundry Layer and undercompensation added
*                   According to Kubo's results
*   01/11/95  hdt     Butterworth filter for torque
*   18/04/96  hdt     Differentiator Filter for acceleration
*   18/04/96  hdt     Kalman filter estimation for torque ripples
*   18/04/96  hdt     H-infinity Controller
*
***** */
#include "DT140X.h"          /* Analog-to-digital converter definitions */
#include <math.h>             /* Sine-wave generator functions */
#include "node_map.h"        /* C30 node address definitions */
#include "stypes.h"          /* Boolean type definition */
#include "servo_refgen.h"    /* Servo reference generation functions */
#include "servo_conv.h"      /* Servo variable value definitions */

#define NUM_CHANNELS 4
#define NUM_MISC_FIELDS 1
#define FIELD_QTY (NUM_CHANNELS+NUM_MISC_FIELDS)
/* Number of fields per sample */
#define NUM_A2D_CHAN    DT1401_A2D_CHANNELS
#define SIGN(X)          ((X) < 0 ? -1 : 1 )

int *time_sample_ptr = (int *) 0xc23000; /* Sample data storage */
short *mbox = (short *) 0xe0; /* Mail boxes */
float *parm_ptr = (float *) 0xc000c0; /* Parameter passing mailboxes */
float *sample_ptr = (float *) 0xc03000; /* Sample data storage */
float Namplitude = 0.0, Vamplitude = 0.0;
float Nbias = 0.0, Vbias = 0.0;
float theta = 0, freq = 0 ; /* Angle and angular frequency of function */
float rise = 0.0;
float fall = 0.0;

```

```

float duty = 1.0;          /* duty cycle ratio for some reference inputs */

void c_int10();
float friction();
float Diff_filter();
float First_order_filter();
float Kalman();
float Control();

/* Constant needed for Subroutines */
int kal_count = 1;
float v_1=0.0, a_1=0.0;
float T_1=0.0, Tf_1=0.0;
float volt2Nm = 6.6, volt2radps = 56.2565;
float J_load = 1.725e-2, Gain = 100;
float T_ripple = 0.0;
float u_1=0,u_2=0,u_3=0,u_4=0,
      y_1=0,y_2=0,y_3=0,y_4=0;

DT140XChan
    str_ch = DT140X_CH3,
    end_ch = DT140X_CH6;
int numchannels = NUM_CHANNELS;          /* number of a2d channels */
int wavetype = 0;                        /* no wave generation by default */
int samplecount = 0;                     /* sample count */
float sample_period = 0.001;             /* Sample period in decimal */
int time_samplecount = 0;

/*----- Timer 1 ETINT1 handler -----*/
void c_int10()
{
    int channel, Nref;
    float a2dsamples[NUM_A2D_CHAN];      /* current set of resolved values */
    float ref_fl, ref_control, Error, ref_fric, ref_total;
    float Tacho, MotorCurrOut, Torque, Torque_f, Torque_clean;
    float vel, acc, T_meas, T_mod, Delta_T, out;

    DT1401Read( str_ch, end_ch, a2dsamples);

    Tacho      = a2dsamples[DT140X_CH4];
    MotorCurrOut = a2dsamples[DT140X_CH5];
    Torque      = a2dsamples[DT140X_CH6];

    vel        = Tacho * volt2radps;      /* Calibration volt --> rad/sec */
    acc        = Diff_filter( vel);        /* Differentiation of velocity */
    T_meas     = Torque * volt2Nm;         /* Calibration volt --> Nm */
    T_mod      = J_load * acc / Gain;      /* Evaluatind Modelled Torque */
    Delta_T    = T_meas - T_mod;           /* Torque Error */
    Torque_f   = T_meas - T_ripple;        /* Kalman Filtered Torque */
    Torque_clean = First_order_filter(Torque_f);
                                           /* Final Filtered Torque */
    T_ripple   = Kalman (Delta_T, vel, sample_period);
                                           /* Finding torque ripple for next
                                           iteration: Perdiction-Type KF */
}

```

```

ref_fl      = reference_generation( wavetype );
Error       = ref_fl - Torque_clean;
ref_control = Control(Error);          /* Generating Control Output */
ref_fric    = friction( ref_control , Tacho );
ref_total   = ref_fric + ref_control;

write_dac0( ref_total );
write_dac1( Torque_clean );

copy(&ref_fl,      sample_ptr++,1,0x0000, 16); /* NRAM->GRAM */
copy(&Tacho,       sample_ptr++,1,0x0000, 16); /* NRAM->GRAM */
copy(&MotorCurrOut, sample_ptr++,1,0x0000, 16); /* NRAM->GRAM */
copy(&Torque_clean, sample_ptr++,1,0x0000, 16); /* NRAM->GRAM */

++samplecount;
enable_intr();
} /* c_int10() */

/* ***** */
void main()
{
    Bool Samples_Ready = True;
    int channel, pause, numsamples;

    copy(parm_ptr,  &Nbias, 1, 0x0000, 21);          /* GRAM->NRAM */
    copy(parm_ptr+2, &sample_period, 1, 0x0000, 21); /* GRAM->NRAM */
    copy(parm_ptr+3, &numsamples, 1, 0x0000, 21);    /* GRAM->NRAM */
    copy(parm_ptr+4, &Namplitude, 1, 0x0000, 21);    /* GRAM->NRAM */
    copy(parm_ptr+5, &wavetype, 1, 0x0000, 21);      /* GRAM->NRAM */
    copy(parm_ptr+6, &freq, 1, 0x0000, 21);          /* GRAM->NRAM */
    copy(parm_ptr+7, &duty, 1, 0x0000, 21);          /* GRAM->NRAM */

    Vbias = PC2V_linear_conv( Nbias );
    Vamplitude = PC2V_linear_conv( Namplitude );
    setup_dac_csr();
    write_dac0( Vbias );

    for (pause=0; pause < 0xffff; pause++);

    /* Initialize a2d converter; sample rate & mapping */
    DT140XInit(str_ch, end_ch);
    t1_schedule_intr(c_int10,sample_period);
    enable_intr();
    t1_start();

    while( samplecount < numsamples ){}
    disable_intr();
    copy(&Samples_Ready,(parm_ptr +1),1,0x0000, 16); /* NRAM->GRAM */
    write_dac0(0.0);
    c30_led_sflash();
}

/* ***** Friction Feedforward ***** */
*
* This routine calculates the Coulomb and viscous friction of the

```

```

* system based on the Identified parameters and adds enough input to
* the system to compensate for that.
*
* Function calls:
*     Controlled_Ref_Signal = friction ( Ref_signal, velocity )
*
* The algorithm includes a treshhold and boundary layer for friction
* at low velocities see Kubo et al. 1996 for details
*
*****/
float friction (inp1, inp2 )          /* Friction Compensation Algorithm */

float inp1, inp2;                     /* inp1: reference signal ;
                                     inp2: Tacho readings      */

{
float Tvp, Tvn, Tsp, Tsn, Tacho_calib_gain,
Torque2volt, inp_comp, Treshhold, under_comp_p, under_comp_n;
Tvp=3.7302e-04;                      /* Viscous friction (+ive velocity direction)*/
Tvn=3.5282e-04;                      /* Viscous friction (-ive velocity direction)*/
Tsp=4.6057e-02;                      /* Static friction (+ive velocity direction) */
Tsn=4.4390e-02;                      /* Static friction (-ive velocity direction) */
Tacho_calib_gain = 56.2565;          /* volt2rad/sec */
                                     /* 1/Km/Servo_Gain relates the compensation torque */
                                     /* to the compensation voltage to be feed to the servo */
Torque2volt      = 70.15 ;

Treshhold = 1.0;                     /* velocity treshhold (rad/sec) */
under_comp_p = 0.95;                 /* Under compensation ratio */
under_comp_n = 0.95;                 /* Under compensation ratio */

inp2= inp2 * Tacho_calib_gain;

if ( inp2 > Treshhold ) /* Positive velocity*/
    inp_comp= under_comp_p * Torque2volt * ( Tvp * inp2 + Tsp );
else if (inp2 < -Treshhold) /* Negative velocity*/
    inp_comp = under_comp_n * Torque2volt * ( Tvn * inp2 - Tsn );
else /* boundary layer routine */
{
    if ( inp1 > 0.0 ) /* Positive reference command */
        inp_comp= under_comp_p * Torque2volt * ( Tvp * inp2 + Tsp );
    else if ( inp1 < 0.0 ) /* Negative reference command */
        inp_comp = under_comp_n * Torque2volt * ( Tvn * inp2 - Tsn );
    }
return( inp1 + inp_comp );
}

```

```

/* ----- Control algoritm, discritized H_infinity controller ----- */

float Control( inp )
float inp;
{
    float out;

/* check file control.c for more info about the controllers */

/* Control4
Ws = (s+300)/5(s+3)    Wu = 2e-3    gamma = 0.7344
*/

float n[5]={
    1.613232636335038e+02,
    -3.070818844474838e+02,
    -1.476520800977480e+01,
    3.070860612409937e+02,
    -1.465538788302189e+02
};

float d[5]={
    1.000000000000000e+00,
    -2.194950808074451e+00,
    9.761822803324164e-01,
    6.340890488734918e-01,
    -4.153191877594238e-01
};

    out=((n[0]*inp+n[1]*u_1+n[2]*u_2+n[3]*u_3+n[4]*u_4)-
        (d[4]*y_4+d[3]*y_3+d[2]*y_2+d[1]*y_1))/d[0];

    u_4 = u_3;  u_3 = u_2;  u_2 = u_1;  u_1 = inp;
    y_4 = y_3;  y_3 = y_2;  y_2 = y_1;  y_1 = out;

    return(out);
}

/***** First order Filter *****/
*
* Discretized Differentiator Filter  $H(s) = 1 / (0.05 s + 1)$ 
* Discretized by tustin method for  $T_s = 1$  ms
*
* Function Calls : Tf = First_order_filter (T)
*
*   input :      T =   noisy signal
*   output:      Tf = filtered signal
*****/

float First_order_filter (T)
float T;
{
    float Tf;

```

```

float B[2]={
    0.00990099009901,
    0.00990099009901,
};
float A[2]={
    1.00000000000000,
    -0.98019801980198
};

/*
float B[2]={
    0.02439024390244,
    0.02439024390244,
};
float A[2]={
    1.00000000000000,
    -0.95121951219512,
};
*/
Tf=B[0]*T + B[1]*T_1 - A[1]*Tf_1;
T_1 = T; Tf_1 = Tf;

return(Tf);
}

/* -- end of file fric_kfdr_hinf.c ---*/

```

5. Open-Loop Frequency Response

5.1. The Host Portion

```

/*-----
*   FILE: load_freq_kfdr.c
*
*   Calling Sequence: Main program.
*
*   Function: Interactive loader for sampling at constant time intervals.
*
*   Revision Information:
*       Date       By   Changes Made....
*       22/06/92   ptd   Modified loadadlights.c
*       24/02/94   ptd   Conversions of sample data from Ti_float format.
*       20/10/95   hdt   Updating the reference signals to incorporate
*                           duty-cycled functions.
*-----*/
#include <sys/errno.h>
#include <stdio.h>                /* contains declaration of scanf() */
#include <stdlib.h>               /* declaration of atof();extern double atof() */
#include <malloc.h>
#include "skyc30v.h"
#include "stypes.h"
#include "ti2ie3cnv.h"

#define NODE1          1                /* C30 node */
#define NODE2          2                /* C30 node */
#define CARD           0                /* the Challenger card number */
#define GLOBAL         0                /* for GRAM */

#define NODE1FILE       "freq_kfdr.out" /* C30 program */
#define NODE2FILE       "c30loop.out"  /* C30 program */
#define GMEM_BASE       0xc00000        /* Global memory */
#define GMEM_OFFSET_PP  0xc0           /* C30 Parameter Passing space */
#define GMEM_OFFSET_SD  0x3000         /* C30 Sample Data space */
#define GMEM_OFFSET_TB  0x23000        /* C30 time sample data */

/*
#define FIELD_QTY        4
#define FIELD_QTY_TOTAL (FIELD_QTY+1)
#define FIELD_QTY_TS     2
*/

extern TiFloat ie3_to_ti();
extern float ti_to_ie3();
SKYC30_device *sky_open_card();

long *samples;
int time_samplecount;
/*----- */

main()
{
    unsigned long i,delay, numsamples;

```

```

    BIT_32 *addr;
    int dummy, status;
    char fname1[80], inputvar[80];
    float sample_period, max_time;
    Bool Samples_Ready = False;
    TiFloat tmpf_Ti, sample_period_Ti, max_time_Ti;
    char *src, *dst;
    SKYC30_device *device;                /* pointer to the hardware */

/* Acquire the card and make sure that the card is not doing anything. */
device = sky_open_card(0);                /* assign the card */
if ( device == NULL )
{
    perror("sky_open_card: ");
    exit(-1);
}
sky_nodes_off(0);                        /* turn it off */

/* Load the C30 main program into node 1. */

status = sky_ld_file (0, NODE1, NODE1FILE);
if (status != 0)
{
    printf ("Node 1, sky_ld_file() for file '%s' failed; status = %d\n",
            fname1, status);
    exit (-1);
}

/* Load the C30 dummy loop program into node 2. */
status = sky_ld_file (0, NODE2, NODE2FILE);
if (status != 0)
{
    printf ("Node 2, sky_ld_file() for file '%s' failed; status = %d\n",
            NODE2FILE, status);
    exit (-1);
}

/* Parameter passing via C30 global memory. */
addr = device->mem + GMEM_OFFSET_PP;
printf("    Enter sample period Ts (s):  ");
scanf("%s",inputvar);
sample_period = atof(inputvar);
tmpf_Ti = ie3_to_ti( &sample_period );
src = (char *) &tmpf_Ti;
dst = (char *) &sample_period_Ti;
*dst++ = *src++; *dst++ = *src++; *dst++ = *src++; *dst++ = *src++;
status = sky_wr_mem (CARD, GLOBAL, addr, 1, &sample_period_Ti);
if (status != 0)
{
    printf ("sky_wr_mem() failed; status = %d\n", status);
    exit (-1);
}

/* Parameter passing via C30 global memory. */
addr = device->mem + GMEM_OFFSET_PP + 1;
status = sky_wr_mem (CARD, GLOBAL, addr, 1, &Samples_Ready);
if (status != 0)

```



```

    {
        printf ("sky_wr_mem() failed; status = %d\n", status);
        exit (-1);
    }

/* Parameter passing via C30 global memory. */
addr = device->mem + GMEM_OFFSET_PP + 2;
printf("    Enter the expected experiment time (s):  ");
scanf("%s",inputvar);
max_time = atof(inputvar);
tmpf_Ti = ie3_to_ti( &max_time);
    src = (char *) &tmpf_Ti;
    dst = (char *) &max_time_Ti;
*dst++ = *src++; *dst++ = *src++; *dst++ = *src++; *dst++ = *src++;
    status = sky_wr_mem (CARD, GLOBAL, addr, 1, &max_time_Ti);
    if (status != 0)
    {
        printf ("sky_wr_mem() failed; status = %d\n", status);
        exit (-1);
    }

/* Run the C30 programs. */
sky_reset_go(0);
addr = device->mem + GMEM_OFFSET_PP + 0x1;
    while( Samples_Ready == False ){                                /* Samples ready */
        printf("    Waiting for Samples_Ready flag...\n");
        for (delay=0; delay< (max_time * 8 /0.0001) ; delay++);
        status = sky_rd_mem(CARD,GLOBAL,addr,1,&Samples_Ready);
        if (status != 0)
        {
            printf("sky_rd_mem failed \n");
            exit (-1);
        }
    }
    printf("    Experiment Closed ... \n
           Check if frequency response is complete! \n");

    samples++;

/* convert hexadecimal short integer to decimal format */
}
/* ----- End of Module ----- */

```

5.2. The C30 Portion

```

/-----
* FILE: freq_kalman.c
*
* Calling Sequence:
*
* Function: 1- C30 node "C" program to respond to C30V timer-generated
*           interrupts using timer 0 and sample a2d inputs and the pulse count
*           from the quadrature decoder.
*           2- Kalman filter is used to filter the torque ripples
*           3- Accomodate Siglab input and output
*
* Revision Information:
*   Date      By    Changes Made....
*
*   26/02/92  ptd   Cleaned up and documented.
*   11/03/92  ptd   Expanded from 2 channels to four.
*   24/02/94  ptd   DT140X analog input boards; float functionalities.
*   12/10/95  hdt   Friction compensation loop added
*   13/11/95  hdt   Modified for frequency response estimates
*   22/04/96  hdt   Differentiator subroutine
*   22/04/96  hdt   Kalman Estimator for torque ripple
*
----- */
#include "DT140X.h"          /* Analog-to-digital converter definitions */
#include <math.h>             /* Sine-wave generator functions */
#include "node_map.h"        /* C30 node address definitions */
#include "stypes.h"          /* Boolean type definition */
#include "servo_refgen.h"    /* Servo reference generation functions */
#include "servo_conv.h"      /* Servo variable value definitions */

#define NUM_CHANNELS 4
#define NUM_MISC_FIELDS 1
/* Number of fields per sample */
#define FIELD_QTY (NUM_CHANNELS+NUM_MISC_FIELDS)
#define NUM_A2D_CHAN DT1401_A2D_CHANNELS
#define SIGN(X) ((X) < 0 ? -1 : 1 )

int *time_sample_ptr = (int *) 0xc23000; /* Sample data storage */
short *mbox = (short *) 0xe0; /* Mail boxes */
float *parm_ptr = (float *) 0xc000c0; /* Parameter passing mailboxes */
float *sample_ptr = (float *) 0xc03000; /* Sample data storage */
/* float Namplitude = 0.0, Vamplitude = 0.0;
float Nbias = 0.0, Vbias = 0.0;
float theta = 0, freq = 0;
float rise = 0.0;
float fall = 0.0;
*/
void c_int10();
float Diff_filter();
float First_order_filter();
float Kalman();

/* Constant needed for Subroutines */
int kal_count = 1;
float v_1=0.0, a_1=0.0;

```

```

float T_1=0.0, Tf_1=0.0;
float volt2Nm = 6.6, volt2radps = 56.2565;
float J_load = 1.725e-2, Gain = 100;
float T_ripple = 0.0;

DT140XChan
str_ch = DT140X_CH3,          /* Start channel of a2d = 3 */
end_ch = DT140X_CH6;         /* End channel of a2d = 6 */
int numchannels = NUM_CHANNELS; /* number of a2d channels */
/* int wavetype = 0; */        /* no wave generation by default */
int samplecount = 0;          /* sample count */
float sample_period = 0.001;   /* Sample period in decimal */
int time_samplecount = 0;

/*----- Timer 1 ETINT1 handler -----*/
void c_int10()
{
    int *t1_counter_reg = (int *) T1_CNTR_REG;
    int channel, timecnt, timemark = 0xffffffff;
    float a2dsamples[NUM_A2D_CHAN]; /* current set of resolved values */
    short pcsample;
    short EncoderPulses;
    int Nref;
    float Tacho, MotorCurrOut, Torque, Torque_f, Torque_clean;
    float vel, acc, T_meas, T_mod, Delta_T, out;

    t1_stop();
    timecnt = *t1_counter_reg;
    t1_start();
    copy( PB_LAT_RD_ADDR, &pcsample, 1, 0xca00, VME_2_NRAM);

    DT1401Read( str_ch, end_ch, a2dsamples);

    Tacho      = a2dsamples[DT140X_CH4];
    MotorCurrOut = a2dsamples[DT140X_CH5];
    Torque      = a2dsamples[DT140X_CH6];

    vel      = Tacho * volt2radps; /* Calibration volt --> rad/sec */
    acc      = Diff_filter( vel); /* Differentiation of velocity */
    T_meas   = Torque * volt2Nm; /* Calibration volt --> Nm */
    T_mod     = J_load * acc / Gain; /* Evaluatind Modelled Torque */
    Delta_T  = T_meas - T_mod; /* Torque Error */
    Torque_f  = T_meas - T_ripple; /* Kalman Filtered Torque */
    Torque_clean = First_order_filter(Torque_f); /* Final Filtered Torque */
    T_ripple  = Kalman (Delta_T, vel, sample_period); /* Finding torque ripple for next
                                                    iteration: Prediction-Type KF */

    out = Torque_clean * 10.0;
    write_dac0(out);

    time_samplecount = time_samplecount + 3;
    ++samplecount;
    enable_intr();
}

```

```

} /* c_int10() */

/* ***** */
void main()
{
    Bool Samples_Ready = True;
    int channel, pause, numsamples;
    float max_time = 300; /* Maximum exp. time initialized to 5 min*/
    float t1period = 514.0; /* max. allowable Timer period in */
                                /* seconds */
/* end declaration addition */

    copy(parm_ptr, &sample_period, 1, 0x0000, GRAM_2_NRAM);
    copy(parm_ptr+2, &max_time, 1, 0x0000, GRAM_2_NRAM);
    numsamples = (int)(max_time / sample_period);
/*
    copy(parm_ptr, &Nbias , 1, 0x0000, GRAM_2_NRAM);
    copy(parm_ptr+4, &Namlitude, 1, 0x0000, GRAM_2_NRAM);
    copy(parm_ptr+5, &wavetype, 1, 0x0000, GRAM_2_NRAM);
    copy(parm_ptr+6, &freq, 1, 0x0000, GRAM_2_NRAM);
*/

/* Initialize the hardware "my addition" */

    xvme085_init(); /* Prototype board encoder interface initialized */
    init_c30v_bus(); /* C30 node and buses initialized */
    c30_led_on(); /* Port configuration reset, mode 0 */
    t0_schedule_intr(c_int10, sample_period); /* CPU interrupt level 0x9 */
    enable_ie_int3(); /* CPU external interrupt #3 enabled */
/*
    Vbias = PC2V_linear_conv( Nbias );
    Vamlitude = PC2V_linear_conv( Namlitude );
    setup_dac_csr();
*/

    write_dac0( 0.0 );

    for (pause=0; pause < 0xffff; pause++);
    /* Initialize a2d converter; sample rate & mapping */
    DT140XInit(str_ch, end_ch);
    t1_schedule_intr(c_int10, sample_period);
    enable_intr();
    t1_start();

    while( samplecount < numsamples ){

        disable_intr();
        copy(&time_samplecount, (parm_ptr + 10), 1, 0x0000, NRAM_2_GRAM);
        copy(&Samples_Ready, (parm_ptr + 1), 1, 0x0000, 16); /* NRAM->GRAM */
        write_dac0(0.0);
        c30_led_sflash();
    }

/* -- end of file freq_kalman.c ---*/

```

6. Closed-Loop Frequency Response

```

/-----
* FILE: freq_fric_hinf.c
*
* Calling Sequence:
*
* Function: 1- C30 node "C" program to respond to C30V timer-generated
*           interrupts using timer 0 and sample a2d inputs and the pulse count
*           from the quadrature decoder.
*           4- Kalman filter is used to filter the torque ripples
*
* Revision Information:
*   Date      By    Changes Made....
*
*   26/02/92  ptd   Cleaned up and documented.
*   11/03/92  ptd   Expanded from 2 channels to four.
*   24/02/94  ptd   DT140X analog input boards; float functionalities.
*   12/10/95  hdt   Friction compensation loop added
*   13/11/95  hdt   Modified for frequency response estimates
*   22/04/96  hdt   Differentiator subroutine
*   22/04/96  hdt   Kalman Estimator for torque ripple
*   02/08/96  hdt   H-infinity torque feedback
*   29/10/96  hdt   Boundry Layer and undercompensation added
*                   According to Kubo's results
*----- */

#include "DT140X.h"          /* Analog-to-digital converter definitions */
#include <math.h>             /* Sine-wave generator functions */
#include "node_map.h"        /* C30 node address definitions */
#include "stypes.h"          /* Boolean type definition */
#include "servo_refgen.h"    /* Servo reference generation functions */
#include "servo_conv.h"      /* Servo variable value definitions */

#define NUM_CHANNELS 4
#define NUM_MISC_FIELDS 1
#define FIELD_QTY (NUM_CHANNELS+NUM_MISC_FIELDS)
#define NUM_A2D_CHAN DT1401_A2D_CHANNELS
#define SIGN(X) ((X) < 0 ? -1 : 1 )

int *time_sample_ptr = (int *) 0xc23000; /* Sample data storage */
short *mbox = (short *) 0xe0; /* Mail boxes */
float *parm_ptr = (float *) 0xc000c0; /* Parameter passing mailboxes */
float *sample_ptr = (float *) 0xc03000; /* Sample data storage */
/* float Namplitude = 0.0, Vamplitude = 0.0;
float Nbias = 0.0, Vbias = 0.0;
float theta = 0, freq = 0 ;
float rise = 0.0;
float fall = 0.0;
*/
void c_int10();
float friction();
float Diff_filter();
float First_order_filter();

```

```

float Kalman();
float Control();

/* Constant needed for Subroutines */
int kal_count = 1;
float v_1=0.0, a_1=0.0;
float T_1=0.0, Tf_1=0.0;
float volt2Nm = 6.6, volt2radps = 56.2565;
float J_load = 1.725e-2, Gain = 100;
float T_ripple = 0.0;
float u_1=0,u_2=0,u_3=0,u_4=0,
      y_1=0,y_2=0,y_3=0,y_4=0;

DT140XChan
str_ch = DT140X_CH3,          /* Start channel of a2d = 3 */
end_ch = DT140X_CH6;         /* End channel of a2d = 6 */
int numchannels = NUM_CHANNELS; /* number of a2d channels */
/* int wavetype = 0; */        /* no wave generation by default */
int samplecount = 0;           /* sample count */
float sample_period = 0.001;   /* Sample period in decimal */
int time_samplecount = 0;

/* Timer 1 ETINT1 handler -----*/
void c_int10()
{
    int *t1_counter_reg = (int *) T1_CNTR_REG;
    int channel, timecnt, timemark = 0xffffffff;
    float a2dsamples[NUM_A2D_CHAN]; /* current set of resolved values */
    short pcsample;
    short EncoderPulses;
    int Nref;
    float Tacho, MotorCurrOut, Torque, Torque_f, Torque_clean, Error;
    float vel, acc, T_meas, T_mod, Delta_T, out;
    float ref_fl, ref_fl_mag, ref_control, ref_fric, ref_total;
    t1_stop();
    timecnt = *t1_counter_reg;
    t1_start();
    copy( PB_LAT_RD_ADDR,&pcsample,1,0xca00,VME_2_NRAM);

    DT1401Read( str_ch, end_ch, a2dsamples);

    ref_fl_mag = a2dsamples[DT140X_CH3];
    Tacho = a2dsamples[DT140X_CH4];
    MotorCurrOut = a2dsamples[DT140X_CH5];
    Torque = a2dsamples[DT140X_CH6];

    vel = Tacho * volt2radps; /* Calibration volt --> rad/sec */
    acc = Diff_filter( vel); /* Differentiation of velocity */
    T_meas = Torque * volt2Nm; /* Calibration volt --> Nm */
    T_mod = J_load * acc / Gain; /* Evaluatind Modelled Torque */
    Delta_T = T_meas - T_mod; /* Torque Error */
    Torque_f = T_meas - T_ripple; /* Kalman Filtered Torque */
    Torque_clean = First_order_filter(Torque_f); /* Final Filtered Torque */
    T_ripple = Kalman (Delta_T, vel, sample_period);

```

```

/* Finding torque ripple for next
   iteration: Prediction-Type KF */

ref_fl = (ref_fl_mag ) / 10.0;      /* attenuate the magnified input */
/* signal to its real value      */

Error      = ref_fl - Torque_clean;
ref_control = Control(Error);        /* Generating Control Output */
ref_fric    = friction( ref_control , Tacho );
ref_total   = ref_fric + ref_control;

write_dac0( ref_total );             /* Write controlled output to D/A*/
out = Torque_clean * 10.0;           /* Magnify Torque signal      */
write_dac1(out);                     /* Write Magnified Torque to D/A */

pcsample &= 0x0000ffff;              /* Avoid hi/lo duplication of short */
EncoderPulses = pcsample;
time_samplecount = time_samplecount + 3;
++samplecount;
enable_intr();

} /* c_int10() */

/* ***** */
void main()
{
    Bool Samples_Ready = True;
    int channel, pause, numsamples;
    float max_time = 300; /* Maximum experiment time initialized to 5 min*/
    float tperiod = 514.0; /* max. allowable Timer period in */
                                /* seconds */

    copy(parm_ptr, &sample_period, 1, 0x0000, GRAM_2_NRAM);
    copy(parm_ptr+2, &max_time, 1, 0x0000, GRAM_2_NRAM);
    numsamples = (int)(max_time / sample_period);

    /* Initialize the hardware */

    xvme085_init(); /* Prototype board encoder interface initialized */
    init_c30v_bus(); /* C30 node and buses initialized */
    c30_led_on(); /* Port configuration reset, mode 0 */
    t0_schedule_intr(c_int10, sample_period); /* CPU interrupt level 0x9 */
    enable_ie_int3(); /* CPU external interrupt #3 enabled */

    write_dac0( 0.0 );

    for (pause=0; pause < 0xffff; pause++);

    /* Initialize a2d converter; sample rate & mapping */
    DT140XInit(str_ch, end_ch);
    t1_schedule_intr(c_int10, sample_period);
    enable_intr();
    t1_start();

    while( samplecount < numsamples ){

        disable_intr();
        copy(&time_samplecount, (parm_ptr + 10), 1, 0x0000, NRAM_2_GRAM);
    }
}

```

```

    copy(&Samples_Ready,(parm_ptr +1),1,0x0000, 16);          /* NRAM->GRAM */
    write_dac0(0.0);
    c30_led_sflash();
}

/***** Friction Feedforward *****/
*
* This routine calculates the Coulomb and viscous friction of the
* system based on the Identified parameters and adds enough input to
* the system to compensate for that.
*
* Function calls:
*     Controlled_Ref_Signal = friction ( Ref_signal, velocity )
*
* The algorithm includes a treshhold and boundary layer for friction
* at low velocities see Kubo et al. 1996 for details
*
*****/

float friction (inp1, inp2 )          /* Friction Compensation Algorithm */

    float inp1, inp2;                  /* inp1: reference signal ;
                                      inp2: Tacho readings      */

{
    float Tvp, Tvn, Tsp, Tsn, Tacho_calib_gain,
    Torque2volt, inp_comp, Treshhold, under_comp_p, under_comp_n;
    Tvp=3.7302e-04;                    /* Viscous friction (+ive velocity direction)*/
    Tvn=3.5282e-04;                    /* Viscous friction (-ive velocity direction)*/
    Tsp=4.6057e-02;                    /* Static friction (+ive velocity direction) */
    Tsn=4.4390e-02;                    /* Static friction (-ive velocity direction) */
    Tacho_calib_gain = 56.2565;        /* volt2rad/sec */
    /* 1/Km/Servo_Gain relates the compensation torque */
    /* to the compensation voltage to be feed to the servo */
    Torque2volt = 70.15 ;

    Treshhold = 1.0;                   /* velocity treshhold (rad/sec) */
    under_comp_p = 0.95;               /* Under compensation ratio */
    under_comp_n = 0.95;               /* Under compensation ratio */

    inp2= inp2 * Tacho_calib_gain;

    if ( inp2 > Treshhold ) /* Positive velocity*/

        inp_comp= under_comp_p * Torque2volt * ( Tvp * inp2 + Tsp );

    else if (inp2 < -Treshhold) /* Negative velocity*/

        inp_comp = under_comp_n * Torque2volt * ( Tvn * inp2 - Tsn );

    else /* boundary layer routine */
    {
        if ( inp1 > 0.0 ) /* Positive reference command */

            inp_comp= under_comp_p * Torque2volt * ( Tvp * inp2 + Tsp );
    }
}

```



```

        else if ( inp1 < 0.0 ) /* Negative reference command */
            inp_comp = under_comp_n * Torque2volt * ( Tvn * inp2 - Tsn );
    }
    return( inp1 + inp_comp );
}

/* ----- Control algoritm, discritized H_infinity controller ----- */

float Control( inp )
float inp;
{
    float out;

    /* check file control.c for more info about the controllers */

    /* Control4
    Ws = (s+300)/5(s+3)    Wu = 2e-3    gamma = 0.7344
    */

    float n[5]={
        1.613232636335038e+02,
        -3.070818844474838e+02,
        -1.476520800977480e+01,
        3.070860612409937e+02,
        -1.465538788302189e+02
    };

    float d[5]={
        1.000000000000000e+00,
        -2.194950808074451e+00,
        9.761822803324164e-01,
        6.340890488734918e-01,
        -4.153191877594238e-01
    };

    out=((n[0]*inp+n[1]*u_1+n[2]*u_2+n[3]*u_3+n[4]*u_4)-
        (d[4]*y_4+d[3]*y_3+d[2]*y_2+d[1]*y_1))/d[0];

    u_4 = u_3; u_3 = u_2; u_2 = u_1; u_1 = inp;
    y_4 = y_3; y_3 = y_2; y_2 = y_1; y_1 = out;

    return(out);
}

/* -- end of file freq_kalman.c --*/

```

APPENDIX C

Modelling Scheme Verification Plots

As a continuation of modelling scheme verification addressed in § 1.3 of Chapter 5, in this appendix some more comparison plots of the simulation and experimental results for the system under constrained and free motion are presented. Here experiments with different current input shape functions, and frequencies are examined and illustrated. In all figures, a solid line represent the filtered measured velocity or torque signals from the experiments, while a dotted line represents the simulation outputs. A fifth-order Butterworth filter, by zero-phase distortion routine is employed to filter the measured signals.

1. System under Constrained-Motion

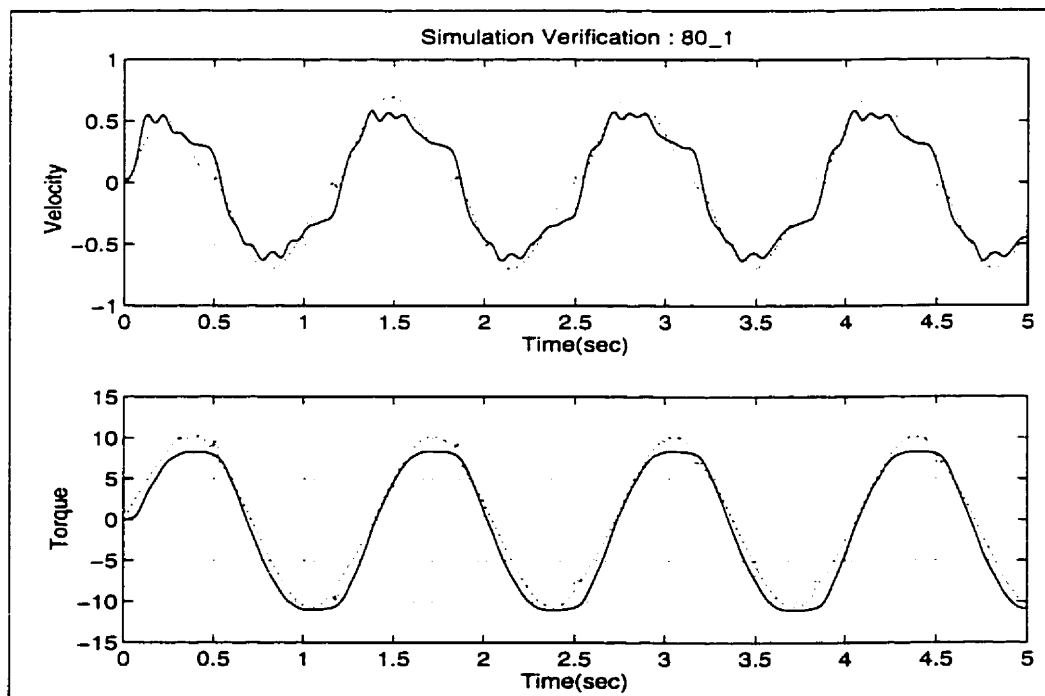


FIGURE C.1. Simulation verification: Sinusoid input, 80% of maximum amplitude.

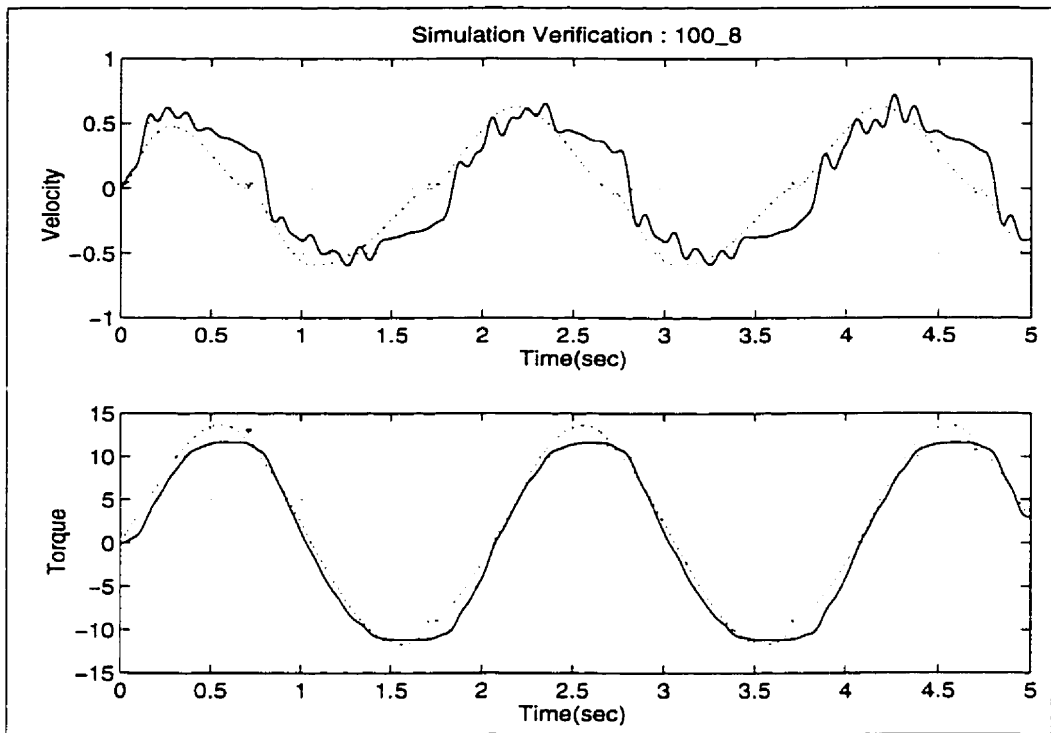


FIGURE C.2. Simulation verification: Asymmetric sinusoid input, with maximum amplitude.

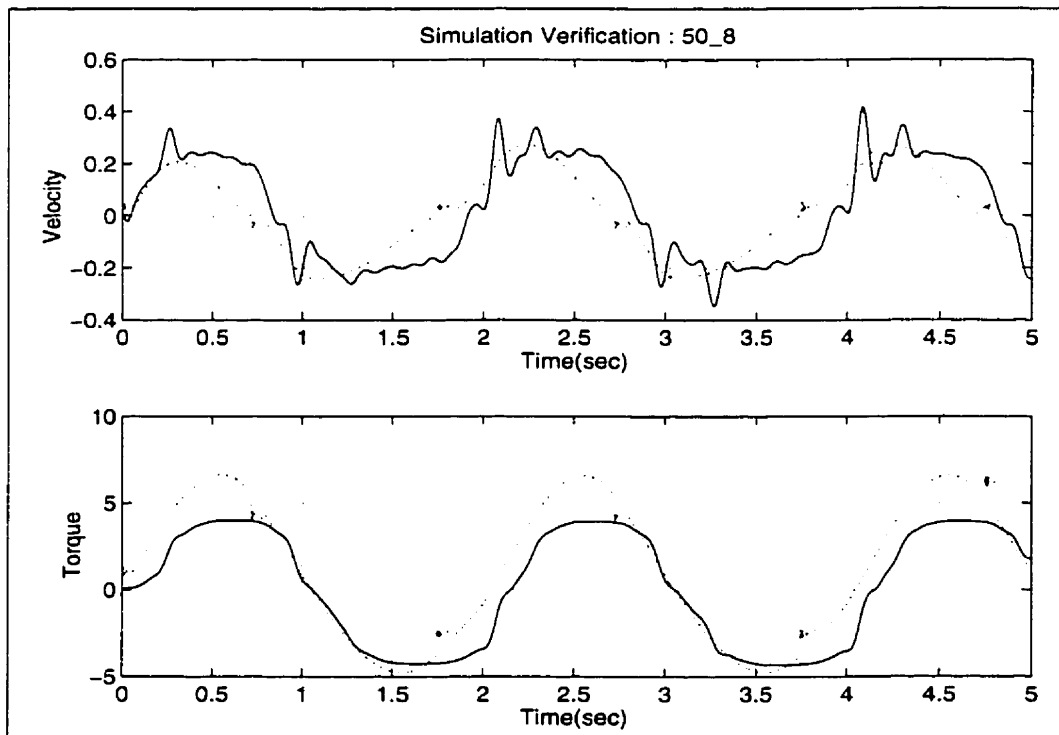


FIGURE C.3. Simulation verification: Asymmetric sinusoid input, 50% of maximum amplitude.

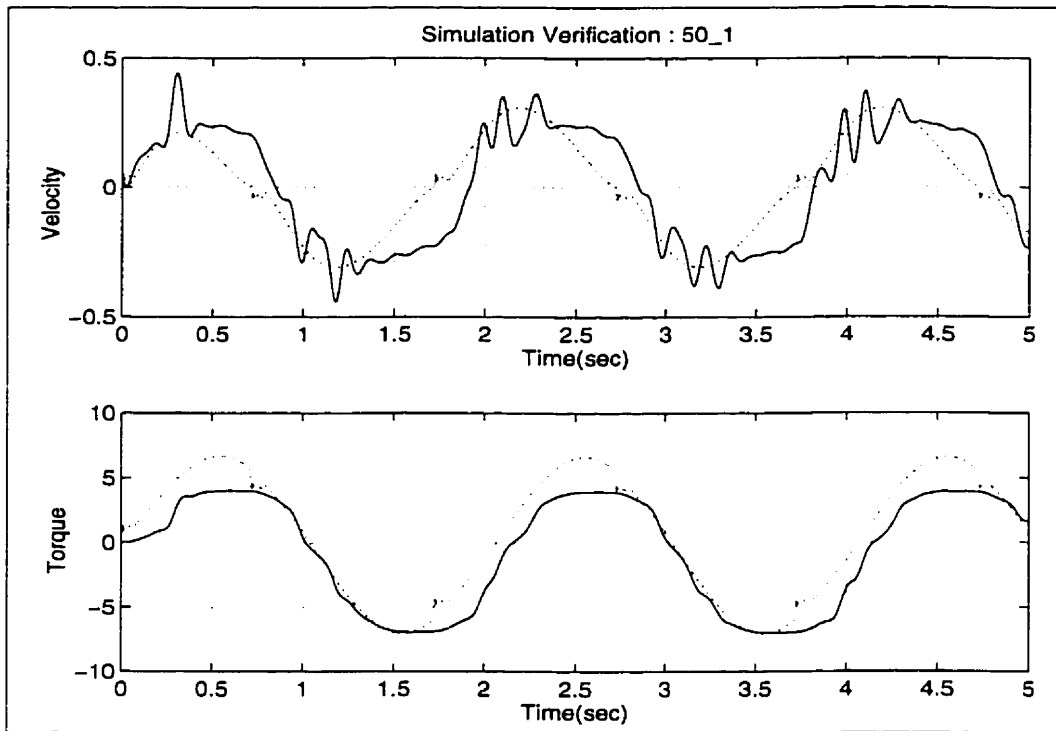


FIGURE C.4. Simulation verification: Sinusoid input, 50% of maximum amplitude.

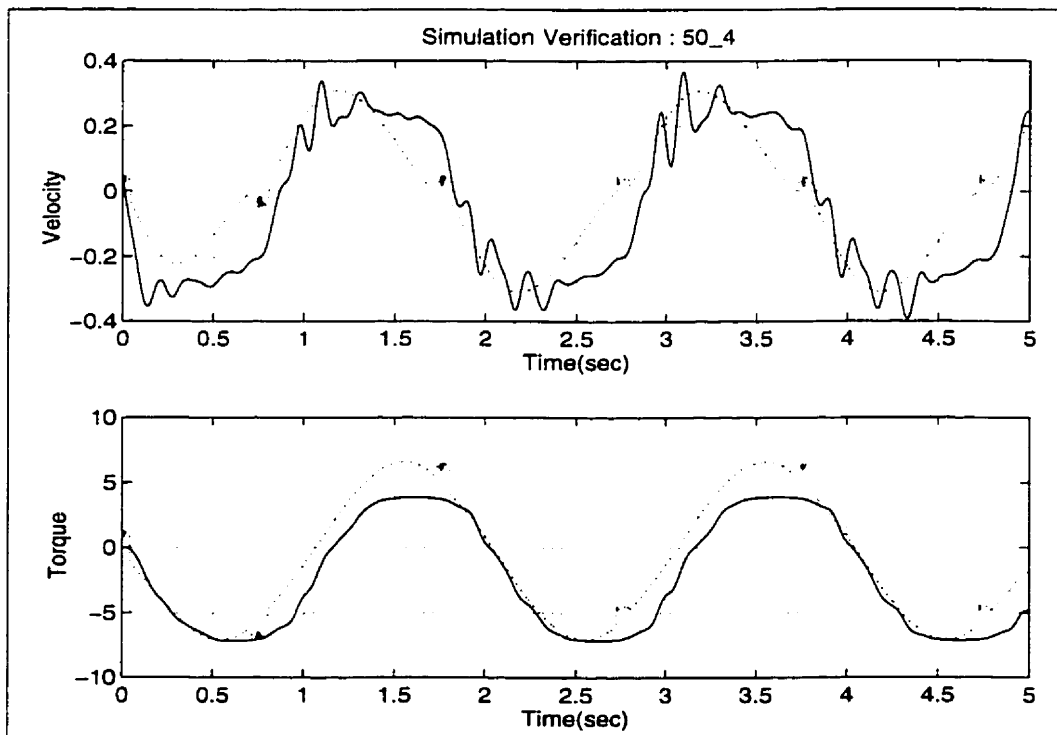


FIGURE C.5. Simulation verification: Inverted sinusoid input, 50% of maximum amplitude.

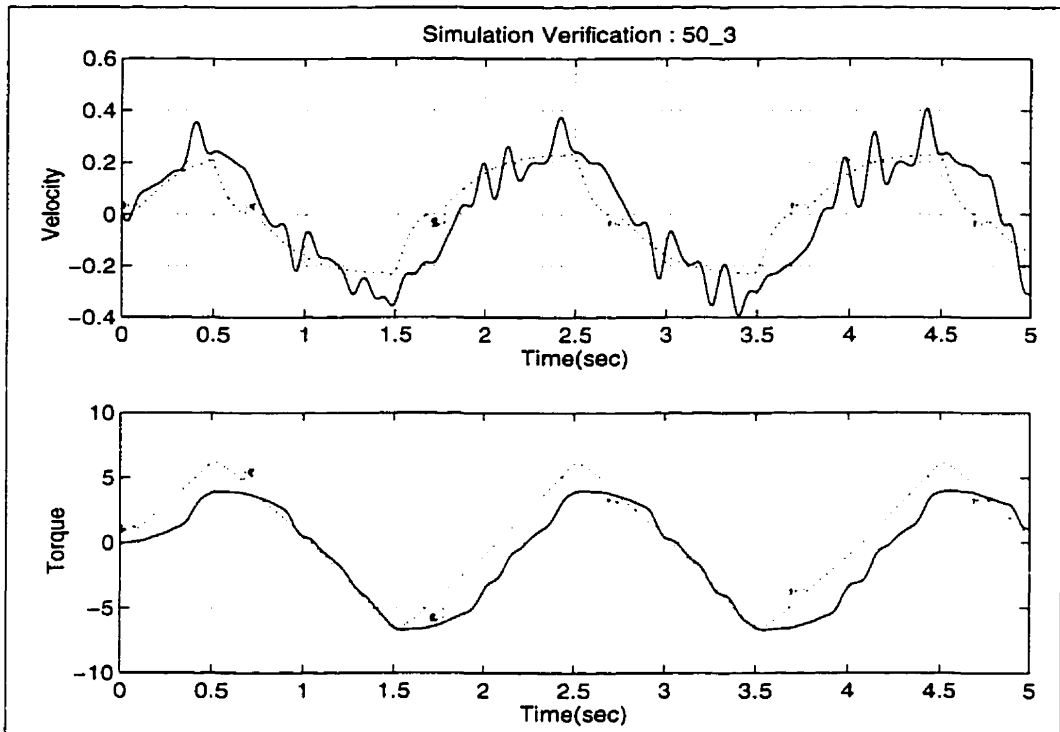


FIGURE C.6. Simulation verification: Triangular input, 50% of maximum amplitude.

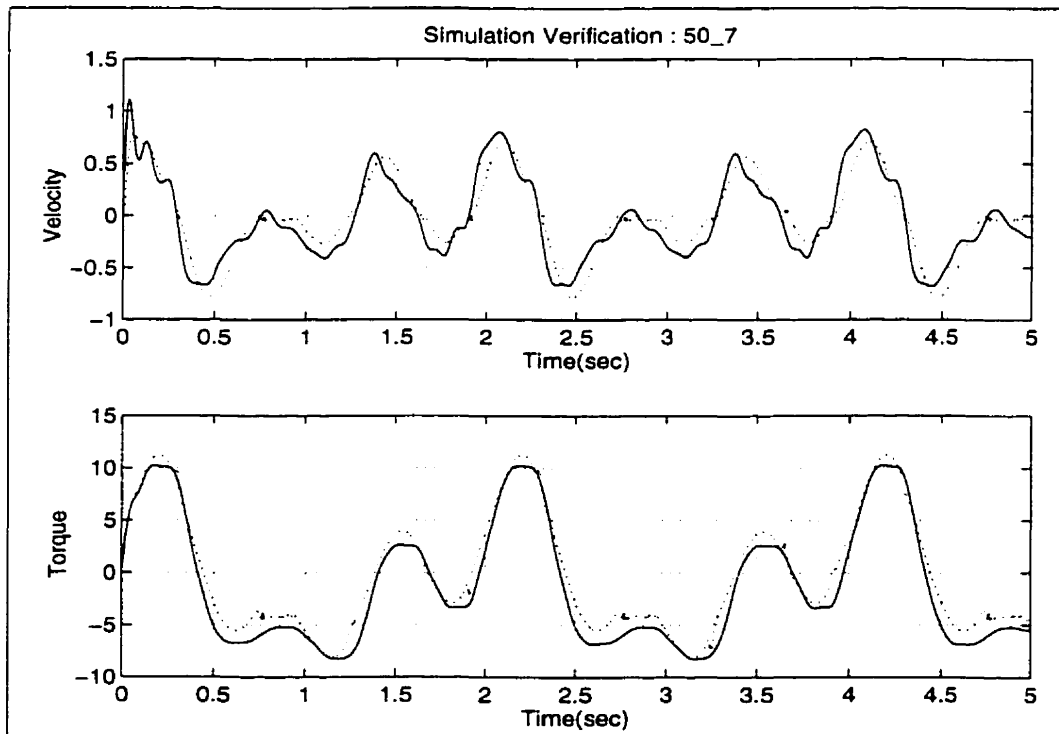


FIGURE C.7. Simulation verification: Composite input, 50% of maximum amplitude.

2. System under Free-Motion

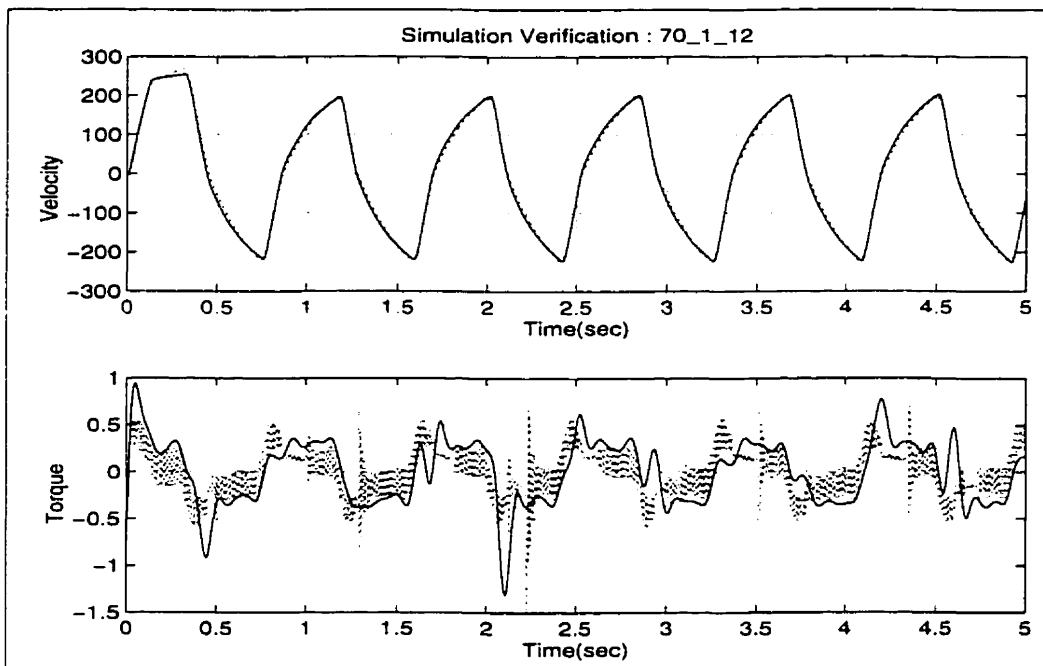


FIGURE C.8. Simulation verification: 1.2 Hz sinusoid input

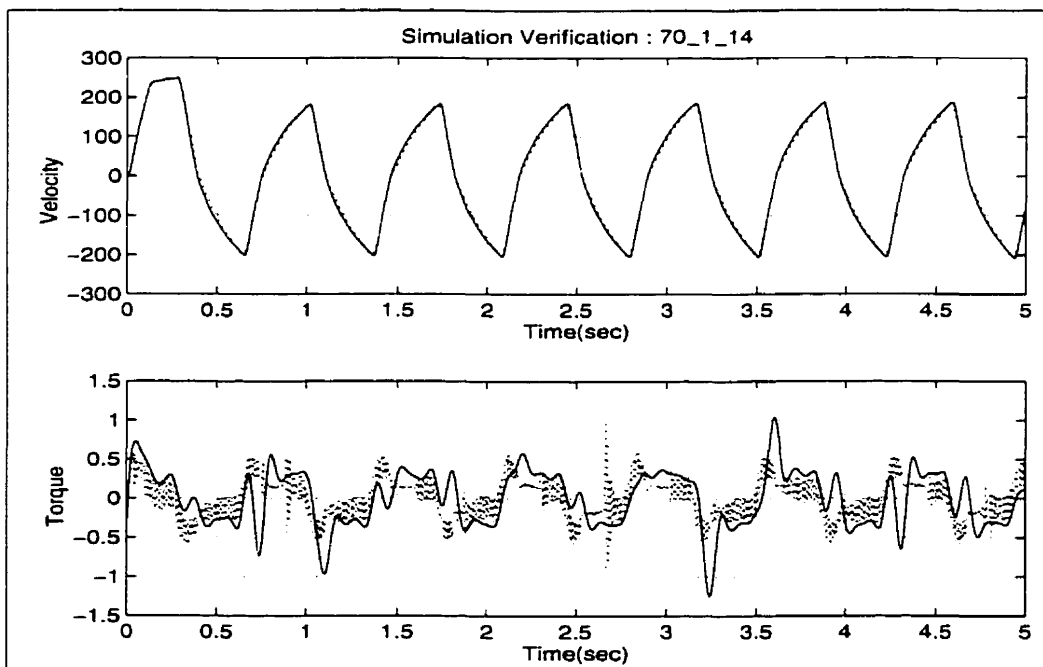


FIGURE C.9. Simulation verification: 1.4 Hz sinusoid input

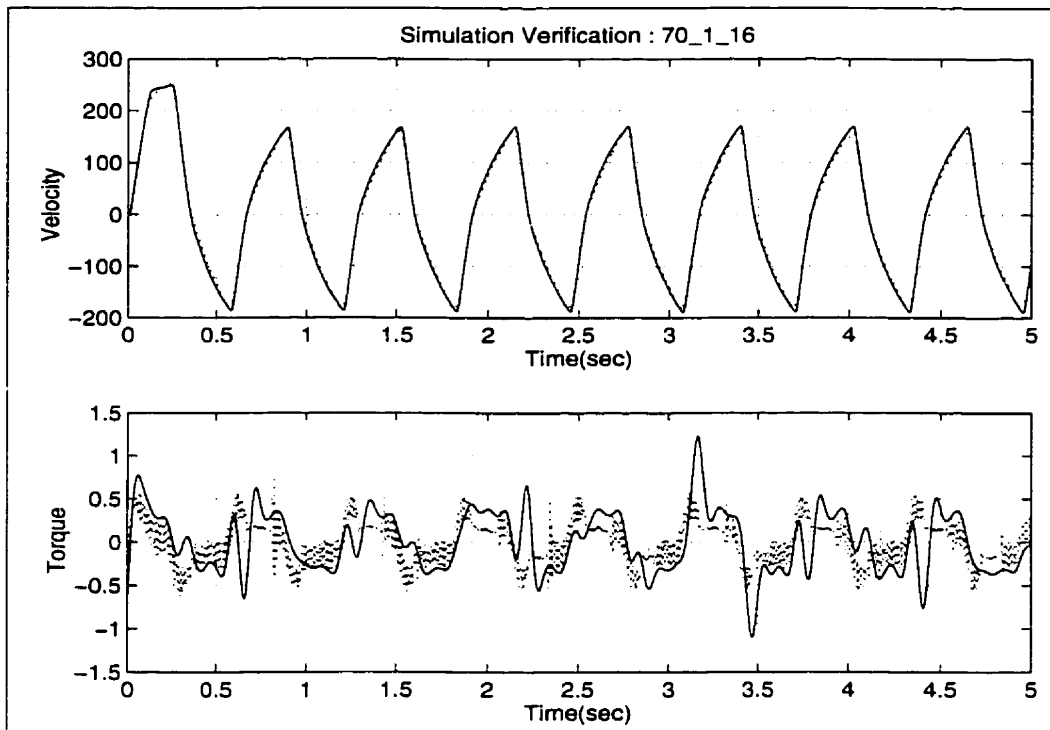


FIGURE C.10. Simulation verification: 1.6 Hz sinusoid input

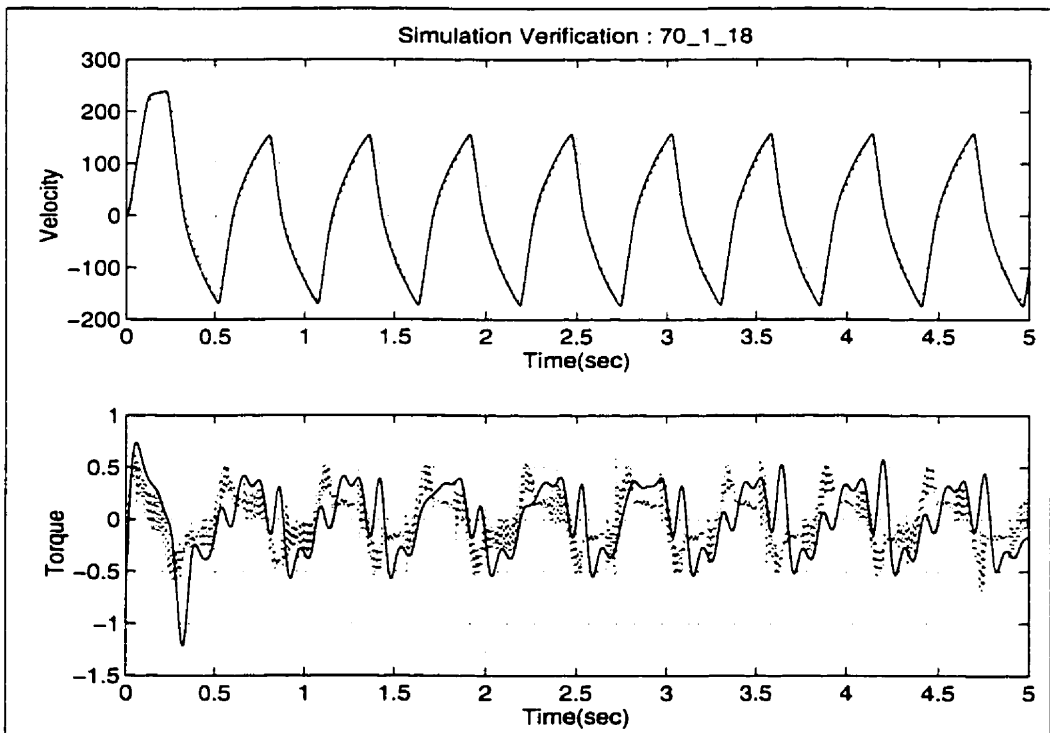


FIGURE C.11. Simulation verification: 1.8 Hz sinusoid input

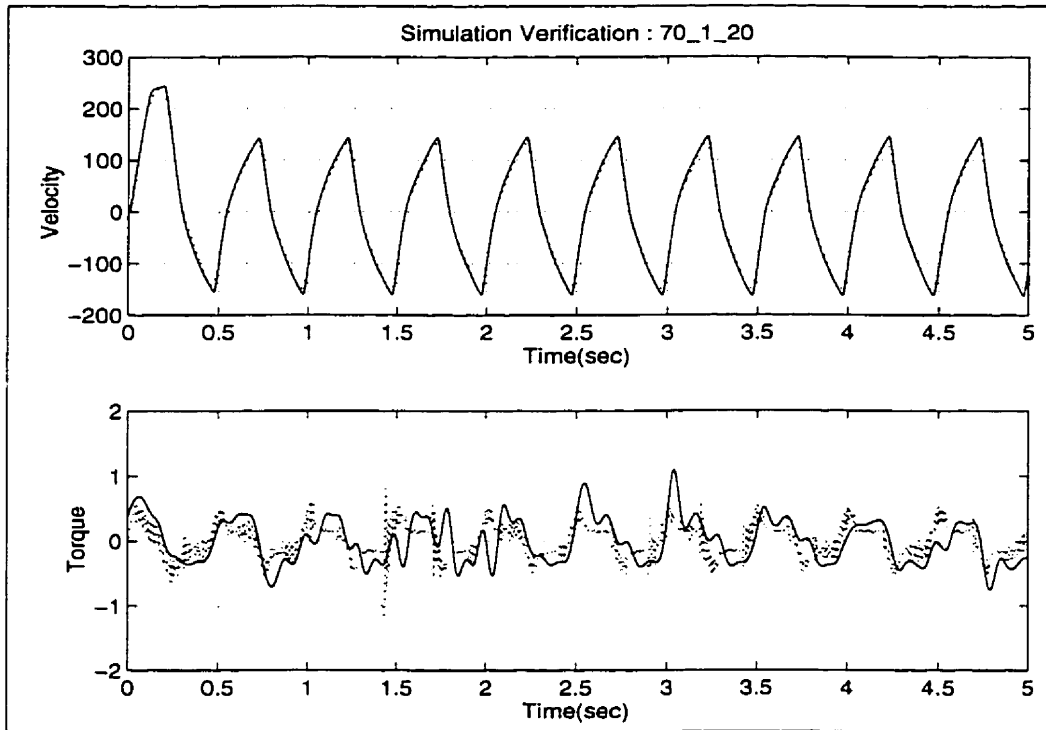


FIGURE C.12. Simulation verification: 2 Hz sinusoid input

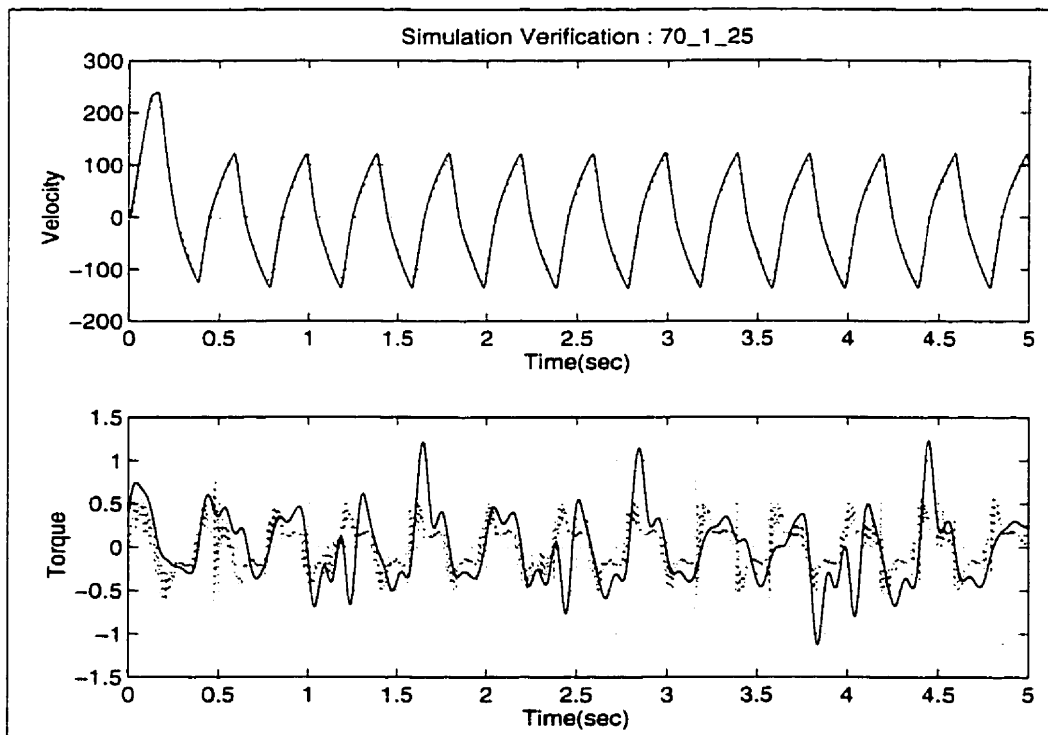


FIGURE C.13. Simulation verification: 2.5 Hz sinusoid input

Document Log:

Manuscript Version 1 — Jan 13, 1997
Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ — 5 March 1997

HAMID D. TAGHIRAD

CENTER FOR INTELLIGENT MACHINES, MCGILL UNIVERSITY, 3480 UNIVERSITY ST., MONTRÉAL (QUÉBEC) H3A
2A7, CANADA, *Tel.* : (514) 398-8207
E-mail address: taghirad@cim.mcgill.ca

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$