

**SATELLITE MOMENTUM WHEEL SPEED CONTROL WITH
DRY ROLLING FRICTION COMPENSATION**

DRAGOS DINCA

Bachelor of Electrical Engineering

Cleveland State University

May, 2004

submitted in partial fulfillment of requirements for the degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

at the

CLEVELAND STATE UNIVERSITY

December, 2006

This thesis has been approved
for the Department of Electrical and Computer Engineering
and the College of Graduate Studies by

Thesis Committee Chairperson, Lili Dong

Department/Date

Taysir H. Nayfeh

Department/Date

Zhiqiang Gao

Department/Date

Majid Rashidi

Department/Date

To my parents and my brother...

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Lili Dong, for her guidance, advice, and support that helped me complete the work of this thesis.

I would like to thank Dr. Taysir Nayfeh, for his guidance and support as well as the opportunity to work at the Industrial Space Systems Lab. The experience accumulated while working at the ISSL is invaluable, as I have also learned to look at engineering from a different perspective.

I would like to thank Dr. Zhiqing Gao, who has been an inspiration during my undergraduate and graduate studies.

I would like to thank Dr. Majid Rashidi, for developing my interest in mechanical engineering while working together on VikSat1.

I would like to thank my colleagues at the ISSL: Brian Fast, Daniel Raible, Andrew Jalics, and Nick Tollis for their help and support.

My deepest appreciation goes to my family and friends for their continuous support, encouragement, and understanding. Most of all, I would like to thank God for all the help and blessings.

SATELLITE MOMENTUM WHEEL SPEED CONTROL WITH DRY ROLLING FRICTION COMPENSATION

DRAGOS DINCA

ABSTRACT

Friction compensation is necessary for many mechanical systems in order to reduce cost, improve safety and increase performance. When problem avoidance techniques are not applicable, a friction compensation controller is used instead. Most friction compensation controllers require an accurate friction model. However, in reality it is very difficult to obtain the accurate model information; consequently the performance of these controllers is limited and dependent on the accuracy of the model. Using a control structure that is based on the form of the problem rather than the specific model coefficients, this thesis presents a new friction compensation technique that is able to estimate and compensate friction in real time. This control structure is robust, less sensitive to the system's dynamic variations, and can be generalized and applied to other systems with different types of friction.

A nonlinear-based speed controller is also proposed, which can be applied to systems with high stiction, limited control effort, or applications that employ a flexible coupling. Hardware implementation results of these controllers are presented as they have been applied to a speed control problem with a high coefficient of dry rolling

friction and other hardware constraints, some of which are: reduced processor bandwidth, limited control effort, and measurement noise. The problem originated from the need of precisely controlling the speed of the momentum wheels for attitude control purposes, an integral part of the first lab-based satellite at Cleveland State University (VikSat1). The engineering design of VikSat1's Attitude Determination and Control Subsystem is also presented.

TABLE OF CONTENTS

	Page
NOMENCLATURE.....	X
LIST OF TABLES	XIII
LIST OF FIGURES	XIV
I. INTRODUCTION.....	1
1.1 Introduction.....	1
1.2 Background.....	4
1.3 Problem Formulation	7
1.4 Thesis Outline	10
II. LITERATURE REVIEW	12
2.1 Understanding Friction	12
2.1.1 The Rolling Element and Rolling Friction.....	14
2.2 Friction Compensation Techniques	17
2.3 Friction Modeling	19
2.4 Reformulating the Friction Problem	23
III. ADCS: DESIGN AND IMPLEMENTATION	25
3.1 From Concept to Reality.....	25
3.2 The ADCS Design Process	28

3.2.1	Drive Electronics	36
3.2.2	Velocity Feedback Sensor.....	37
3.2.3	Onboard Processor	40
IV.	CONTROLLER DESIGN	44
4.1	The Controlled System	45
4.2	Active Disturbance Rejection Control	49
4.2.1	The 1 st Order LADRC.....	51
4.3	Incremental Control	56
4.4	Nonlinear Incremental Control	58
4.4.1	Nonlinear Gains	59
V.	HARDWARE IMPLEMENTATION	63
5.1	Controller Implementation Issues	63
5.1.1	Encoder – Microcontroller Interface.....	64
5.1.2	Sensor Noise	66
5.1.3	Motion Profiles	68
5.1.3	Data Acquisition	70
5.2	Hardware Results	72
5.2.1	No Load Hardware Results – Without External Disturbance.....	73
5.2.2	No Load Hardware Results – With External Disturbance	78

5.2.3	Under Load Hardware Results – Without External Disturbance..	80
5.2.4	Under Load Hardware Results – With External Disturbance.....	83
5.3	Controller Comparison.....	85
VI.	CONCLUDING REMARKS.....	89
6.1	Future Research	90
	REFERENCES.....	92
	APPENDICES	97
A.	Momentum Wheel Motor Datasheet.....	98
B.	Solar Panels Control PCB	99
C.	VikSat1	99
D.	Drive Electronics PCB – Design Schematics	100
E.	ADCS Microcontroller PCB – Design Schematics	102

NOMENCLATURE

AC:	Alternative Current
ADC:	Analog-to-Digital Converter
ADCS:	Attitude Determination and Control Subsystem
ADRC:	Active Disturbance Rejection Control
BDC:	Brush DC
BLDC:	Brushless DC
C&DH:	Command & Data Handling
CPR:	Cycles-Per-Revolution
CSU:	Cleveland State University
DAC:	Digital-to-Analog Converter
DC:	Direct Current
DSP:	Digital Signal Processing
DTOC:	Discrete Time Optimal Control
EMF:	Electromotive Force
ESO:	Extended State Observer
FPGA:	Field Programmable Logic Devices
I/O:	Input-Output Pins

IAC:	Indirect Adaptive Control
IC:	Integrated Circuits
ICM:	Input Capture Module
ISSL:	Industrial Space Systems Lab
JTC:	Joint Torque Control
LADRC:	Linear Active Disturbance Rejection Control
LC:	Learning Control
LED:	Light Emitting Diode
LESO:	Linear Extended State Observer
MIMO:	Multi-Input Multi-Output
MRAC:	Model Reference Adaptive Control
NESO:	Nonlinear Extended State Observer
NIC:	Nonlinear Incremental Control
NPID:	Nonlinear Proportional-Integral-Derivative
PCB:	Printed Circuit Board
PID:	Proportional-Integral-Derivative
PSD:	Position Sensitive Detector
RMS:	Root-Mean-Square
RPM:	Revolutions-Per-Minute

SPI:	Serial Peripheral Interface
TOC:	Time Optimal Control
TTL:	Transistor-Transistor Logic
UART:	Universal Asynchronous Receiver Transmitter
UIO:	Unknown Input Observer

LIST OF TABLES

Table		Page
TABLE I:	Motor Type Selection: Advantages and Disadvantages [33], [34].....	32
TABLE II:	Controller Comparison (Under Load).....	88

LIST OF FIGURES

Figure		Page
Figure 1:	The Momentum Wheel Assembly	5
Figure 2:	Press-Fitting of The Ball Bearings.....	6
Figure 3:	Leonardo da Vinci’s Proposed Ball Bearing Design [15]	15
Figure 4:	Examples of Static Friction Models [20]	21
Figure 5:	VikSat1’s Reference Axes	28
Figure 6:	Electric Power Distribution – Graphical Illustration	34
Figure 7:	Total Power Delivered to the Motor	36
Figure 8:	Drive Electronics PCB	37
Figure 9:	Feedback Encoder	39
Figure 10:	ADCS – Block Diagram	41
Figure 11:	ADCS Main Board.....	43
Figure 12:	Drive Electronics – Block Diagram.....	45
Figure 13:	Motor Model – Block Diagram.....	47
Figure 14:	Mechanical Aspect of the Model – Block Diagram.....	48
Figure 15:	Complete Momentum Wheel Model – Block Diagram.....	48
Figure 16:	Representation of z	56

Figure 17:	Incremental Control – Block Diagram Form	57
Figure 18:	NIC – Graphical Illustration ($\beta = 1$)	61
Figure 19:	NIC – Graphical Illustration ($\beta = 0.1$).....	61
Figure 20:	Encoder Signal – With Load at High Speeds.....	67
Figure 21:	Improved Encoder Signal	68
Figure 22:	Feedback Based Motion Profile – Block Diagram	69
Figure 23:	Pre-filter Profile – 1 st Order	70
Figure 24:	Pre-filter Profile – 2 nd Order	70
Figure 25:	Data Acquisition – Block Diagram.....	72
Figure 26:	LADRC – Simulation Set-up	74
Figure 27:	LADRC – Front End Control Law.....	76
Figure 28:	LADRC – Front End Control Law (Reduced Block Diagram)	76
Figure 29:	Incremental Controller – No Load (Without Disturbance).....	77
Figure 30:	LADRC – No Load (Without Disturbance).....	77
Figure 31:	NIC – No Load (Without Disturbance)	78
Figure 32:	Incremental Controller – No Load (With Disturbance).....	79
Figure 33:	LADRC – No Load (With Disturbance).....	79
Figure 34:	NIC – No Load (With Disturbance).....	80
Figure 35:	Incremental Controller – Under Load (Without Disturbance).....	82

Figure 36:	LADRC – Under Load (Without Disturbance).....	82
Figure 37:	NIC – Under Load (Without Disturbance)	83
Figure 38:	Incremental Controller – Under Load (With Disturbance).....	84
Figure 39:	LADRC – Under Load (With Disturbance).....	84
Figure 40:	NIC – Under Load (With Disturbance)	85

CHAPTER I

INTRODUCTION

1.1 Introduction

The application of friction dates back to prehistory and is one of the decisive elements that have enabled the survival of the human species, as frictional heat was discovered and used to light fires [1]. It was not long until humans developed engineering skills, which enabled them to overcome or use friction, i.e. rolling friction, to their advantage. For example the Egyptians used rollers, wheels and sleds along with lubrication (water, animal fat, mud or oil) to transport objects and build different structures [2]. Nowadays, friction is ever present in mechanical systems and some use it in order to function properly (car speedometers, brakes etc.). Most times, however, it is not desired because it decreases a system's performance by affecting its safety, quality and economic cost [3]. At fault is the nonlinear nature of friction because it induces limit cycles, steady state errors, time delays [3], and stick-slip motions [4]. Therefore, in order

to improve a controlled system's performance, alleviating these frictional disturbances is essential.

One such example, where friction compensation was necessary, is the motivation of this thesis. VikSat1 is the first lab based satellite (its components emulate a real satellite) at Cleveland State University (CSU), and a friction related problem occurred during the manufacturing process of one of its subsystems, the Attitude Determination and Control Subsystem (ADCS). One of the main tasks within the ADCS consists of controlling the momentum wheel speed, but due to the manufacturing problem, speed control with dry rolling friction compensation was required in order for the ADCS to operate properly. In addition to friction, the difficulty of the control problem was elevated due to electrical and mechanical hardware constraints such as reduced processor bandwidth, limited control effort, and measurement noise.

Although problem avoidance is one of the most widely used methods to eliminate frictional disturbances, where hardware change or system redesign is usually implemented, it was not a solution in this case. Thus, it was up to the feedback controller to minimize the unwanted frictional phenomena and improve the subsystem's performance. Most such controllers, however, require a complete friction model. Since models are only approximations of the actual frictional behavior, the problem with model-based friction compensation controllers is their performance limitation and dependency on the model's accuracy. The work of this thesis brings a new idea to the subject of friction compensation: treating friction as another disturbance to the system's desired response, within the framework of a control structure that uses the form of the problem for control purposes rather than the specific model coefficients [47], [49], [52],

[53], yields a robust friction compensation controller that is not dependent on a friction model.

While the results herein are in regard to a speed control application requiring the compensation of a high coefficient of dry rolling friction, this control structure can be generalized and thus applied to other systems with different types of friction. In addition, it can also be used to obtain, otherwise unknown, information about the nonlinear frictional behavior. Depending on the specific application, this information can be used to improve the controller's performance and thus better compensate for the friction phenomena.

In order to deal with the high coefficient of static friction, which became a predominant issue in controlling the angular velocity of the momentum wheels, a new nonlinear based speed controller was also developed. This new controller provides an alternative solution to applications such as this, and/or applications with flexible coupling where a gradual (soft) start is required. At the same time, this controller has good disturbance rejection capabilities, it is easy to implement, and its tuning is intuitive. The nonlinear based controller is a modification to Incremental Control, the first control structure used for momentum wheel speed regulation. For a complete controller comparison, hardware implementation results offer insight and comparison information for all three controllers implemented.

In addition to the contribution to the field of friction compensation, the work of this thesis also includes the successful electrical engineering design of the ADCS. It was during the manufacturing process of this subsystem that the friction problem was identified and a solution was found. Thus, the complete subsystem design (from concept

to the actual realized hardware) is also discussed, along with solutions to the electrical and mechanical constraints, and limitations encountered during the design process.

1.2 Background

Subsystems are a satellite's building blocks and they serve two purposes: first, they ensure the proper functionality of the spacecraft; second, the number and type of these subsystems are determined by the mission's objective and payload. All spacecrafts have a way of knowing their orientation in space. Also, in case the satellite needs to change its current orientation or in case the payload must be reoriented, the satellite must be able to change its direction. Usually, all spacecrafts do have both of these capabilities and in most cases these tasks are lumped into one block, the ADCS [31].

The ADCS onboard VikSat1 determines and controls (stabilizes and orients) the satellite's attitude or orientation, by utilizing two momentum wheels that rotate in opposite directions at speeds of up to 2000 revolutions-per-minute (RPM). The rotation speed of the momentum wheels was chosen based on their inertia and the torque required to rotate the satellite within a predefined frame of time. The difference in angular momentum between the two wheels, which is caused by their respective angular velocity difference, translates to a change in the satellite's angular momentum with respect to its longitudinal axis. The angular momentum difference between the wheels is accomplished by keeping one wheel at a constant speed, while varying the other wheel's angular velocity.

The mechanical aspect of the momentum wheel assembly, which is illustrated in Figure 1, was designed by faculty and students in mechanical engineering¹. Since the wheels' speed varies from 0 to 2000 RPM, a bearing system that would meet these requirements was selected. Bearing systems of different types were considered during the selection process and some of the options included magnetic, gas, air, and rolling ball bearings of the sealed and non-sealed type. Although the goal was to select a bearing system with minimum friction, the magnetic, gas, and air types were not selected because of funding limitations. Also, it was determined that for a low power and low torque application, such as VikSat1, the static friction coefficient under no radial load would be less for a non-sealed ball bearing system than for a sealed one. Thus, a non-sealed rolling ball bearing system was used for each individual momentum wheel.

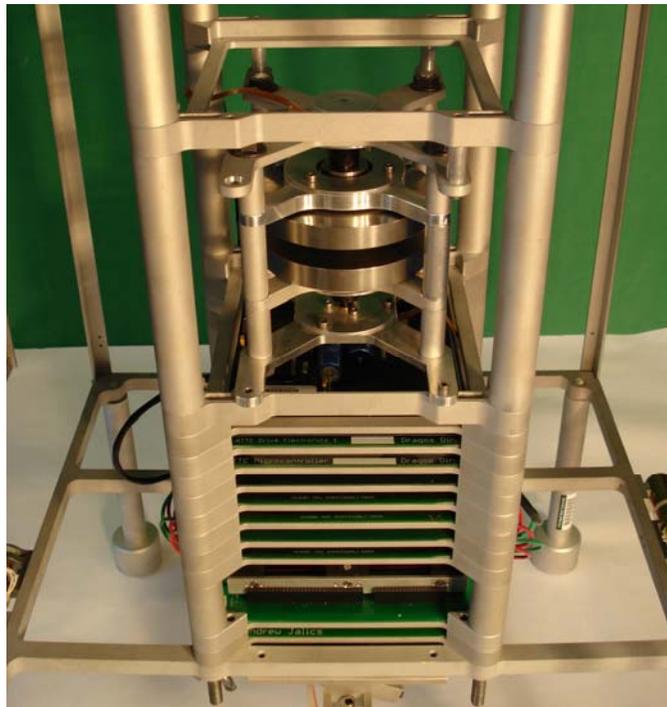


Figure 1: The Momentum Wheel Assembly

¹ Dr. Majid Rashidi, Mechanical Engineering Professor at Cleveland State University, et al.

The rolling element bearing was selected with the mindset of minimizing the momentum wheels' frictional losses. However, during the manufacturing process, a problem occurred when the ball bearings and the momentum wheel shaft were press-fitted (see Figure 2), as more pressure was exerted on the bearings due to the slightly larger momentum wheel shaft diameter (manufacturing imperfections). The additional bearing pressure altered the performance of the momentum wheel assembly by significantly increasing the coefficient of dry rolling friction and adding unknown nonlinearities to the system.



Figure 2: Press-Fitting of The Ball Bearings

1.3 Problem Formulation

Since the bearing problem appeared after the actual design and during the manufacturing process, the concept of design for control, also known as problem avoidance, was no longer applicable. According to [5], the first step in solving the friction related problem would have been to re-design the subsystem or change the hardware. In this case, however, subsystem re-design was not an option due to funding limitations. In addition, that would have required other subsystems to be re-designed or modified also, which would have prolonged the satellite's completion time.

Given that the mechanical hardware could not be re-designed, changing the hardware or acquiring larger motors was not an option either. The momentum wheel motors were sized based on the momentum wheel's inertia (see Chapter 3 for additional information), but most importantly motor sizing was highly influenced by VikSat1's limited electric power budget. Hence, motor replacement was not an option due to the subsystem's electric power and size limitations.

One other alternative solution, in an attempt to avoid the friction problem, would have been to add lubrication to the ball bearings. However, since the momentum wheels utilized a non-sealed bearing system, any lubricant would have attracted dust or other impediments. With time, the friction coefficient would actually increase and the problem would worsen. For that reason, lubrication was not added and now it was up to the speed controller to compensate for the undesired friction. Thus, the challenge was to design and implement a robust speed controller that would precisely control the angular velocity

of the momentum wheels in the presence of dry rolling friction and additional nonlinearities created by extra pressure on the ball bearings.

Furthermore, as a result of the additional pressure exerted on the ball bearings, the sliding friction between the rollers and the solid surfaces was more predominant. That significantly increased stiction or the coefficient of static friction, which meant that the breakaway force had to be much greater. This is a major issue because the motors were originally sized for a much lower friction coefficient. In addition to the high friction coefficient, hardware constraints (electrical and mechanical) made momentum wheel speed control yet more difficult.

With regard to the friction problem, controller design was a difficult task for two reasons. First, the high friction coefficient amplified the nonlinearities present in the system. Second, the motors now deemed to be undersized for this application and thus had limited control effort; that meant that the actuator, the momentum wheel motor in this case, had limited range. In other words, the actuator's response to the input control signal was limited, as a large change in the control signal was not possible and not desired due to the motor current limit and the flexible coupler that used to connect the momentum wheel to the motor (the coupler was observed slipping during certain accelerations). Although more control effort was needed to compensate for the additional friction nonlinearities, a fast changing control signal rate was not possible. In control systems, this scenario is highly undesired because it slows down the actuator response time (also known as actuator time delay).

Hardware constraints (electrical and mechanical) also affected the controller design process. One such constraint was the limited processing bandwidth provided by

the ADCS on board controller. A 16-bit microcontroller was used for all tasks required by the ADCS. Although the dsPIC30F6010 is a high performance microcontroller, its bandwidth was limited due to its interface with the feedback device, and its ability to process variables of the floating point type.

An optical encoder was used to measure the angular velocity of the momentum wheels. While the encoder provides good velocity measurement resolution, as opposed to a tachometer, its pulse measurement significantly reduced the microcontroller's bandwidth. Section 3.2 provides additional encoder information, while Section 5.1.1 describes the constraints imposed by the interface between the processor and the encoder feedback. The second bandwidth constraint was due to the microcontroller's limited ability to calculate floating point type numbers. Since the microcontroller does not have true floating point capabilities, such as a dedicated hardware module, a C-complier was used to emulate floating point numbers instead. Therefore, the available control rate was constrained by the time frame that it took to measure the actual pulse width, and the microcontroller's processing time of the controller equations, and any other additional programming routines.

In most control systems, measurement noise is inevitable and the aptitude of a controller to cope with noise reflects one aspect of its robustness. The momentum wheel speed measurement was also affected by noise, which was induced from two different sources. First, the encoder interface electronics induced voltage spikes on top of the rising and falling edges of the encoder signal. In addition, the rotational motion of the momentum wheels and the interconnecting mechanical components caused phantom like

pulses, known as ringing. These voltage spikes led to false microcontroller triggering, which in turn affected the speed measurement accuracy.

In summary, controlling the angular velocity of the momentum wheels was a difficult control problem due to the following constraints:

- High dry rolling friction coefficient
- Limited actuator range
- Actuator time delay
- Reduced processor bandwidth
- Measurement noise

1.4 Thesis Outline

The remaining chapters provide supporting information and solutions to the above problems. At the same time, this research lays the groundwork for a new framework towards the development of a generalized friction compensation controller. Therefore, in order to better understand the problem at hand, Chapter 2 contains the necessary literature review regarding frictional phenomena (general information, models, and identification techniques), and friction compensation methods. Also in this chapter, the problem is reformulated in the disturbance rejection framework. The ADCS design process is presented in Chapter 3. Chapter 4 discusses the development of the friction compensation controllers implemented in hardware. The corresponding results are

presented in Chapter 5, while Chapter 6 provides concluding remarks and ideas for future research.

CHAPTER II

LITERATURE REVIEW

The friction compensation domain is vast and interdisciplinary. An introduction to friction but more specifically rolling friction is discussed in Section 2.1. A brief survey regarding friction compensation controllers and techniques is included in Section 2.2. Most of these techniques, however, are heavily dependent on the accuracy of the frictional model. Conversely, the research focus has been shifted towards the development of accurate friction models that better describe the frictional behavior. A brief summary of these models is presented in Section 2.3. Section 2.4 provides an alternative solution as the problem is re-formulated in the disturbance rejection paradigm.

2.1 Understanding Friction

The word friction, according to the Merriam's Webster Dictionary, is derived from the Latin word *fricare* (to rub) and is defined as "the force that resists relative motion between two bodies in contact" [6]; examples of such contacts may be between a solid and another solid, a gas or a liquid [7]. Although we might not pay much attention,

friction is ever present in our daily life. Gemant explains that it would be hard to imagine a frictionless process in nature or industry [8]. There are countless examples to defend that statement, as friction is responsible for many of our daily activities. Walking, for example, would not be possible if the coefficient of friction would be less. People would have to use suction pads or other devices in order to walk. If the coefficient would be higher, however, walking would be quite difficult also [9]. Moreover, without it starting, moving or stopping the movement of a body would not be possible [7].

The performance of mechanical systems is reduced in the presence of undesirable friction, which in turn has a direct impact on the economic price tag [3]. In order to decrease this cost, the friction effect can be reduced by design and/or control [10]. System design or re-design, also known as problem avoidance, is usually implemented first [5], and such techniques have been improved greatly by incorporating the vast knowledge of tribology (“the science of rubbing interfaces” [11]), and using the various tools available from different engineering and science disciplines. In addition to system design, friction reduction (if desired) can also be achieved by means of control. Friction compensation is well known throughout literature, and it refers to the ability of a control system to minimize the decreased system performance due to friction through control action [3]. The best method for friction compensation, however, is accomplished by using both system design and control. Thus, the concept of *design for control* should be a guideline when friction compensation is necessary [5].

Friction alone, when undesired, is considered a menace by control systems engineers because it is highly nonlinear and limits the performance of controlled systems by inducing limit cycles, steady state errors, actuator time delays [3], and stick-slip

motions [4]. These nonlinearities are also present in the momentum wheel assembly because of the manufacturing problem, which added unwanted complexity to the system. Moreover, nonlinearities such as these, have affected all kinds of systems for centuries and because of that, the frictional phenomena was considered and studied by great thinkers such as Hero, Galileo, Leonardo da Vinci, Hooke, Newton, Euler, Columb, Raynolds and others [9], [15].

2.1.1 The Rolling Element and Rolling Friction

Going back to prehistory again, in addition to discovering fire, humans have made another significant invention that is related to friction. The rolling element of friction was discovered back then, and humans took full advantage of it as they realized that it is easier to move heavy objects when rollers are used instead of sliding [7]. Hence, it was not long until the wheel was invented almost 5000 years ago (3200 B.C.). It is interesting to note that both writing and the wheel emerged at the same time in Mesopotamia [12]. The rolling element of friction, although people did not know it at that time, played an important role in the development of civilizations.

Rolling friction was extensively used in history prior to and after the invention of the wheel, but it was not truly examined and studied until the works of Leonardo da Vinci. It was the year of 1508 when the first experiment on sliding friction was conducted by da Vinci, and the research following that experiment revealed for the first time the concept of a rolling friction coefficient [13]. In addition, it must be mentioned that it was the great genius, Leonardo da Vinci, who came up with the basic design of the

current ball bearings that is still used today [14]. His work is documented in the Codex Madrid I [20], and in this document it is revealed that he was made aware of the “roller-disc” bearing idea by his assistant, a German mechanic named Giulio [15]. Nonetheless, the remarkable insight for the need to separate the rolling elements is impressive. Figure 3 shows, the ball bearing design proposed by Leonardo da Vinci [12].

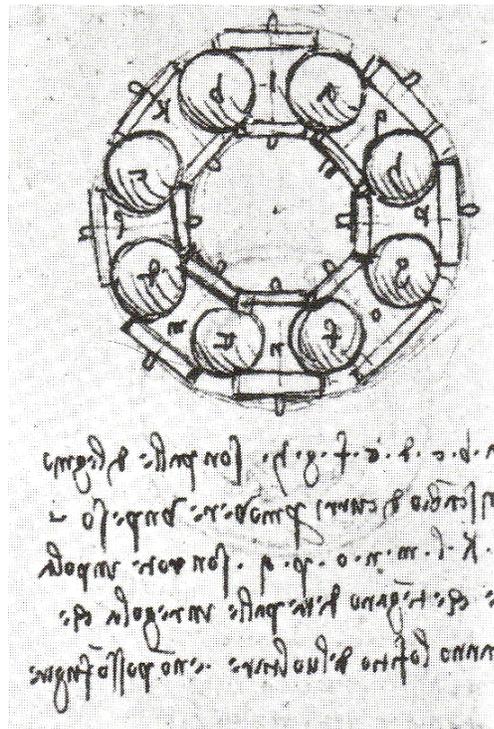


Figure 3: Leonardo da Vinci's Proposed Ball Bearing Design [15]

Da Vinci's concept of rolling friction coefficient was later developed, studied and introduced in the mechanical processes by Delagir, Parin, Euler, Lesley, and others. It was Coulomb, however, who introduced the hypothesis that the nature of friction during rolling is dynamic. This meant that rolling friction is much more complex than originally thought. Additional work by Reynolds in 1876 suggested that the rolling friction force might be due to the sliding or slipping of surfaces in contact [13]. The hypothesis proved

to be correct, as sliding motion creates most of the friction force in roller bearings [5]. In addition to sliding friction, Tomlinson [16] and Tabor [17] showed that atomic and molecular interaction between two bodies mount to additional losses in rolling friction. Thus, according to [7], [13], [14], and [18], rolling frictional losses that occur in bearing systems consist of the following:

1) Sliding friction losses:

- “Sliding in rolling element-raceway contacts due to geometry of contacting surfaces” [14].
- “Sliding due to deformation of contact elements (Heathcote slip)” [14].
- “Sliding between cage pockets and rolling elements” [14].
- “In roller bearings, sliding between roller ends and guiding inner- and/or outer-ring flanges” [14].
- “Slipping due to the ball spinning” [18].

2) Hysteresis losses, caused by the damping capacity of the raceway contacts and the ball bearing materials [18].

3) Microslip at the interface between two bodies that do not have the same elastic properties [7].

4) Surface roughness may affect the coefficient of static friction [7].

The momentum wheel frictional losses are due to the sliding and slipping of the bearings in the rolling element-raceway, which are being amplified by the additional pressure exerted on the bearings.

2.2 Friction Compensation Techniques

Frictional disturbances deteriorate the performance of systems with mechanical components. As cost is a direct consequence, alleviating these types of disturbances is necessary. The techniques used to compensate for friction can be divided into the following categories: problem avoidance, model-based and nonmodel-based compensation techniques [19]. Problem avoidance, or design for control, was briefly explained in the Section 2.1, and its objective is to avoid the control problem by system design, hardware change or by selecting different materials or lubricants [5]. Problem avoidance is the most commonly used friction compensation technique, but according to [19], it does have a higher price tag. Feedback control, however, can reduce this cost and maintain or improve the system's performance. Friction compensation techniques are widely scattered through literature and a brief summary is presented next. No detailed information is included, since that is beyond the scope of this thesis.

In terms of feedback control, the most widely used non-model based control structure is the Proportional-Integral-Derivative (PID) and variations of it such as PI or PD. Integral control is known for its ability to reduce steady state errors and modifications have been made to it in order to improve its efficiency. Modifications such as *position-error dead band* and *lead-lag compensation* are used to reduce hunting (a type of limit cycle) around the set point and other nonlinear effects. Moreover, *integrator resetting* and multiplying the integrator term by the sign of velocity are also used to avoid integrator wind-up, even when velocity changes direction. In addition to integral control, Stiff PD control is used to avoid stick-slip motions and frictional memory (the lag

observed between velocity and the friction force once stiction is overcome). A more advanced form of PID control incorporates *inverse describing functions* by introducing nonlinear functions into the proportional, integral and derivative gains: $f_P(e)$, $f_I(e)$ and $f_D(e)$ [5], [19].

Learning Control (LC), also known as fuzzy logic, is another non-model based control structure; however, this controller is more of a table of corrections. During a specific motion, these corrections are used to adjust the control signal in order to obtain better precision [19]. In addition to learning control, Joint Torque Control (JTC) is a technique that uses a high gain inner torque loop to compensate for friction in actuator-transmission systems. It is also used in applications that require precise control of transmission flexibilities, gravitational loads and others (see [5] for more details). Dither, part of the non-model based techniques, uses a high frequency signal that is introduced into the control signal in order to overcome stiction [19]. Another technique that is similar to dither is the Impulsive Control, which uses a series of small impacts to perform the desired motion [5]. Additional model independent techniques include artificial intelligence, artificial neural networks, and others.

Model based compensation techniques are implemented when an accurate friction model is available. These techniques use feed-forward and a reference model in the control law in order to adjust the controller gains or certain model parameters. Adaptive control and variations of it (Indirect Adaptive Control and Direct Adaptive Control) are widely used because they can adapt to changes in the friction model. The Indirect Adaptive Control (IAC) estimates the friction model parameters online (using observers or other types of estimators or filters) and its advantage consists of additional supervisory

loops that are used in order to verify the estimated parameters before they are used in the control law. There are other variations of IAC in the literature, and in most cases these variations are categorized as Observer Based Compensation Techniques. The Direct Adaptive Control, on the other hand, does not use on-line parameter estimation for the friction model; instead, the controller gains are adapted, based on a specific model, in order to minimize the tracking error. An example of Direct Adaptive Control is the Model Reference Adaptive Control (MRAC), which is based on Lyapunov functions [5], [19].

In conclusion, the effectiveness of non-model based methods is reduced due to cost, as these techniques require significant effort and time to develop and implement. One other disadvantage is the constant tuning required by these techniques, as the system parameters change with time, temperature and other factors. Some of the non-model based methods still use a linear model for controller design purposes and constant tuning becomes a serious cost constraint. Regarding the model-based friction compensation controllers, their performance is dependent on the accuracy of the friction model. Thus, an introduction to friction modeling is presented in the next section.

2.3 Friction Modeling

Since most friction compensation controllers are highly dependent on an accurate friction model, the research has been focused on developing friction identification techniques and more accurate or descriptive friction models. There are many friction

models in the literature, and according to [3] they can be divided into two categories: static and dynamic. The static or classical models attempt to model the most obvious aspects of friction:

- Static friction or stiction, opposed to dynamic friction, is the friction force at rest that keeps an object from moving [3]. A very important observation was made in [3], where it is suggested that the friction force at rest is higher than the Coulomb friction.
- Coulomb friction, a constant friction force that opposes motion with a magnitude that is independent of velocity and contact area [3].
- Viscous friction represents a component of the friction force that is usually associated with Coulomb friction; its magnitude increases linearly with increasing velocity [20].

Therefore, as illustrated in Figure 4, classical models consist of any of these frictional behaviors or combinations of them. The disadvantage with these models is that velocity at zero cannot be detected. Karnopp attempted to fix this problem, as he developed a new model (the Karnopp model) that is able to detect the zero velocity point. The Karnopp model helped differentiate the equations between sticking and sliding, but the zero velocity interval still does not truly reflect real friction. Another improved classical model is that of Armstrong's, which attempts to model some aspects of dynamic friction. Unlike the other models where velocity is assumed to be constant, Armstrong's model offers insight into the frictional phenomena as velocity is varied [3].

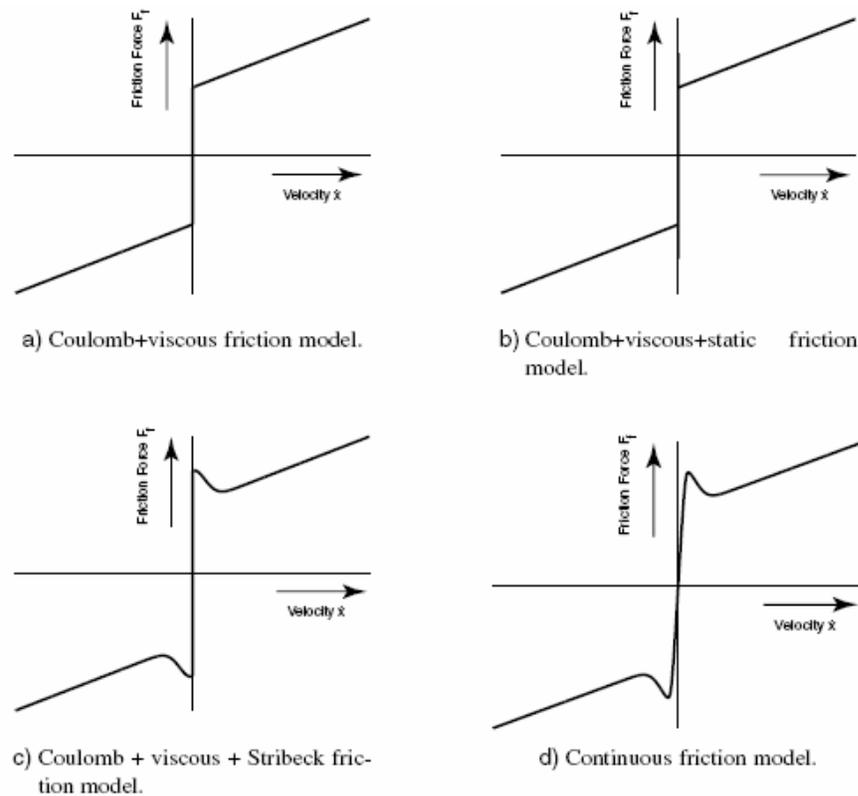


Figure 4: Examples of Static Friction Models [20]

Static models offer a glimpse of the true frictional behavior; however, a dynamic model is required in order to fully describe rolling friction. Dynamic models are more realistic as they include time dependencies and additional frictional phenomena such as:

- Presliding displacement, defined as a spring-like behavior due to limited stiffness of contact asperities [20], is actually the displacement of rolling and sliding contacts prior to true sliding [19].
- Rising static friction, the transition from not-sliding to sliding, is the force required to overcome static friction (the break-away force) and initiate motion. This force varies with the increase of applied force and with time spent at zero velocity (dwell time) [19], [20].

- Stribeck Friction or the Stribeck Curve occurs in most lubricated and some dry contacts [19]; it is defined as a continuous decrease in the friction force at low velocities [20].
- Frictional Lag or Frictional Memory is the lag observed when a change in velocity or load does not correspond to a similar change in the friction force [19]. In other words, the frictional force is larger for increasing velocities than for decreasing velocities. This dynamic behavior becomes more obvious for large accelerations and/or decelerations [20].

The Dahl model is used to describe rolling friction in ball bearing systems. This model was the result of friction experiments in servo systems with ball bearings, and it is one of the first dynamic models developed (P. R. Dahl introduced it in 1966) [3]. The original Dahl model does not capture all frictional phenomena such as the Stribeck effect or stiction, and modifications have been made to it over time in order to include other nonlinearities (improved models are described in [21], [22], [23] and [24]). One such modification is the Bliman-Sorine model [23], which uses two Dahl models in parallel to account for stiction [3]. Another modification is the LuGre model [24], which uses a velocity varying coefficient to represent static friction [3]. Moreover, in addition to stiction, the LuGre model is able to describe the following nonlinearities: the Stribeck effect, frictional lag, hysteresis and stick-slip motions [20]. In order to accurately capture the real frictional phenomena in rolling friction, additional improvements have been made to both models. That however, leads to increased model complexity which makes controller design and analysis much more difficult [20]. There are other dynamic models

in the literature, which are not included here due to the scope and the problem addressed in this thesis.

2.4 Reformulating the Friction Problem

Once problem avoidance techniques are not a viable solution, feedback control is employed in order to compensate for friction. As explained in Section 2.2, cost is a rigorous constraint with non-model based control structures, as they are empirically developed and tuned. Still, most friction compensation controllers do require an accurate friction model. However, “a detailed friction model is often difficult to obtain” [19] because friction is highly nonlinear and time variant. As the actual model parameters change with time, the efficiency of these controllers is questionable as their performance is limited and dependent on the accuracy of the model [19], [47].

The proposed solution brought fourth by this thesis is based on a control structure that is less sensitive to system parameter variations, by treating friction as an external disturbance to the desired system response. In addition, using the form of the problem rather than the specific coefficients of the system model makes this control structure ideal for applications that require a robust friction compensation controller. This control structure is known as Active Disturbance Rejection Control (ADRC) [47], [49], [52], [53], and its development as well as application to the momentum wheel speed control problem is discussed in Section 4.2. It follows that a friction related control problem can

now be reformulated, within the framework of ADRC, to be a disturbance rejection control problem.

CHAPTER III

ADCS: DESIGN AND IMPLEMENTATION

The overall design of the ADCS, from concept to the actual realized hardware, was successful. Although the design is complex with many components (electrical and mechanical), this Chapter provides only the high level documentation of the subsystem's design and implementation process.

3.1 From Concept to Reality

The ADCS is considered to be “the most complex and least intuitive of the spacecraft vehicle design disciplines” [31], where often times this subsystem dictates the general design of the spacecraft. Cost and time constraints, as it was the case with VikSat1, often elevate the subsystem's level of complexity.

In order to design any system, one must fully research and understand its basic functionality. Such was the case with VikSat1 as the ADCS design required the understanding of its basic terminology. Thus, the term *attitude* refers to the spacecraft's

orientation in space [28]. Obtaining the spacecraft's attitude, relative to an inertial reference frame (inertial space) or to an external reference frame, i.e., a specific body (the Earth, the Sun, the stars or a local magnetic field) [28] is known as *attitude determination*. Different sensors are used to measure the attitude, where the measurement's accuracy is dependent on the type and quality of these sensors. Gyroscopes, for example, measure the centrifugal acceleration of a spacecraft's inertial frame; this information is vital in determining whether the spacecraft has changed its original orientation. Other sensors that measure the attitude with respect to an external reference frame include earth horizon sensors, sun sensors, star sensors, magnetometers, and others. These sensors provide only a unit direction vector reference since they do not measure the actual distance. VikaSat1 incorporates a sun sensor for attitude determination.

As the name implies, attitude determination constitutes only one portion of the ADCS. The other part, attitude control, must first maintain the current spacecraft orientation, a process known as attitude stabilization, and second it must be able to orient or reorient the spacecraft in space to a different attitude (attitude maneuver) [28]. These tasks must be performed in the presence of external disturbance forces or torques (aerodynamic, gravity-gradient or magnetic), and a variety of miscellaneous disturbances such as solar radiation pressure or others that may be internally generated during the course of the spacecraft's mission [31].

In order to achieve attitude stabilization and attitude maneuver, spacecrafts incorporate either passive or active attitude control methods. Passive attitude control relies purely on the environment to generate the necessary control torques. The

advantage consists of longer spacecraft lifetime since a limited number of moving components are utilized with this method. The disadvantage, however, is that it does not guarantee stability and has limited accuracy. Active attitude control, on the other hand, is more accurate, and can reach the desired control objectives faster and with more flexibility because it uses feedback control concepts, where sensors determine the necessary corrective inertial torques and the onboard actuators implement them [31]. Examples of such actuators include gas jets, magnetic coils, reaction/momentum wheels, ion or electric thrusters. Efficiency is a major constraint with this method because it adds complexity to the system, as it requires a power source, materials with limited lifetime and, most often, ground control [28].

An active attitude control system is implemented onboard VikSat1, where a momentum wheel assembly is used to generate the corrective torques that are required to stabilize and orient the satellite. The assembly is comprised of two flywheels, feedback devices, and motors along with drive and control electronics which are used, within the spacecraft's control loop, to drive both wheels to their respective desired speeds. Given that VikSat1 is the first generation lab satellite at CSU, the momentum wheel assembly was designed to control VikSat1's attitude around its yaw axis only (see Figure 5). Since momentum wheels are reaction wheels that operate at higher speeds, a more complex inner velocity control loop is required in order to maintain and adjust each wheel's angular velocity. The advantage of using momentum wheels is the relative fast response. However, the disadvantage consists of added complexity to the ADCS because they are heavy, costly, the design involves moving components, and the controls are cumbersome [28], [30], [31].

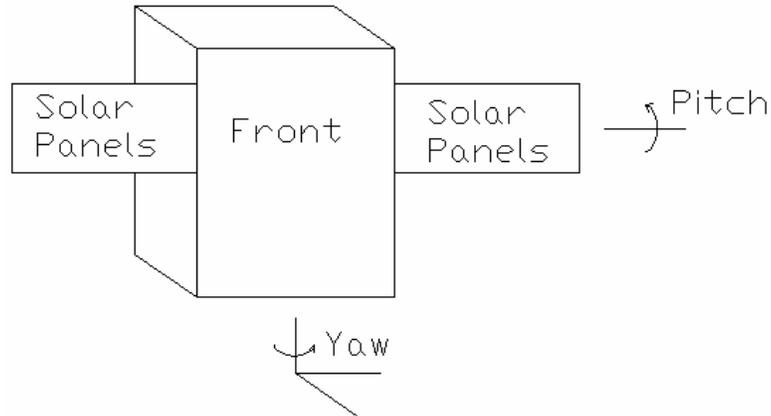


Figure 5: VikSat1's Reference Axes

In order to achieve attitude stabilization and attitude maneuvering the two momentum wheels counter-rotate at speeds of up to 2000 RPM. Under normal operation, the control system was designed such that attitude stabilization is achieved during the zero momentum bias operating point or when both momentum wheels have the same speed. Whenever an attitude change is required, the angular velocity of one wheel is increased or decreased while the other is held constant. The speed difference between the two wheels creates a momentum difference that is translated to the satellite's angular momentum, which in turn orients the satellite.

3.2 The ADCS Design Process

VikSat1's main requirement was to rotate 180° in one minute. To make this a reality, the ADCS design began with sizing the momentum wheels. The torque necessary to overcome VikSat1's angular momentum was the basis for the momentum wheel

design, since the wheels' momentum difference would create that torque. In order to calculate the satellite's rotation torque, Dr. Rashidi and a team of students from his senior design class had to first determine the satellite's moment of inertia about its longitudinal axis. The inertia of the satellite was relatively easy to calculate since VikSat1 is considered to be a solid cylinder. In addition, special care was taken to include the moment of inertia created by the length and weight of its solar panels. Knowing the satellite's inertia and the desired angular acceleration, which was based on VikSat1's requirement (180° in one minute), final calculations showed that the momentum wheels had to generate a torque equal or greater than $\tau_{sat} = 3.84 \times 10^{-3} \text{ N} \cdot \text{m}$.

Since the torque required to rotate the satellite was the basis for the momentum wheel design, equation (3.1) was used to calculate the wheel's inertia

$$J_{wheel} = \frac{\tau_{sat}}{\alpha_{wheel}} \quad (3.1)$$

where α_{wheel} is the angular acceleration of the momentum wheel over a predefined period of time, as given by (3.2). Recall that attitude control is achieved when one wheel is held constant while the speed of the other is varied:

$$\alpha_{wheel} = \frac{|\omega_{final} - \omega_{initial}|}{t_{accel}} \quad (3.2)$$

Equations (3.1) and (3.2) were used to determine the inertia of the momentum wheel. That value was then used to determine the wheel's design parameters such as mass, diameter, and thickness.

In order to connect the wheel to the motor, a small wheel shaft was designed to withstand the torques and speeds provided by the motor. The actual connection between the wheel shaft and the motor shaft was accomplished through a flexible coupler. It was decided that a flexible bellows coupler would be the appropriate choice in order to compensate for planar, angular and axial misalignments. Once the momentum wheels were designed, the stainless steel alloy was the material chosen for their fabrication.

The mechanical aspect of the ADCS was completed with the casing design and manufacture of the momentum wheel assembly, and the selection of a proper bearing system. As explained in Section 1.2, there are different types of bearing systems that could have been used for this application: magnetic, gas, air, and sealed/non-sealed sliding or rolling type of bearing systems. Taking into account that VikSat1 is only a lab prototype model, funding restrictions limited the choice of a desired bearing system such as magnetic, gas or air. Choosing between the sealed and the non-sealed type, it was decided that the non-sealed type would minimize the coefficient of static friction for a low torque and low power application. In addition, for the same considerations and after careful calculations that took into account the radial and axial load stress as well as the weight that had to be supported, rolling bearings were chosen over the sliding type. Therefore, a non-sealed rolling ball bearing system was selected and implemented.

In order to bring to life the mechanical design, the ADCS electrical design includes software along with hardware components, which ensures the subsystem's proper functionality. The electrical hardware consists of the momentum wheel motors along with the corresponding drive and control electronics, feedback devices, digital-to-analog (DAC) and analog-to-digital (ADC) converters, a microcontroller, communication

and control Integrated Circuits (ICs), and other necessary electronics. Three Printed Circuit Boards (PCBs), one for each main ADCS functionality, were thus designed: microcontroller processing, drive electronics (momentum wheel motors), and solar panels (board layout was done at the ISSL²).

The momentum wheel actuator or the momentum wheel motor is one of the main components of the ADCS. Proper motor selection and sizing was required in order to ensure that the momentum wheel motor offered the necessary reliability and performance for in-space attitude control. The types of electric motors available for aerospace applications can be either AC (alternative current) or DC (direct current). Since the electric power source available onboard VikSat1 to all subsystems is strictly DC, the types of motors that were considered as actuators for the momentum wheels were Stepper Motors, Brush DC (BDC) Motors and Brushless DC (BLDC) Motors.

The BDC Motors are a popular choice because of their simplicity and relative simple drive electronics. As an example, one can even use a BDC motor with no drive electronics if a simple open-loop configuration is desired. Their biggest disadvantage, however, is that the motor windings are energized through the electrical connection made between the commutator and brushes. This sliding electrical connection affects the motor's lifespan because the brushes wear out with time. The brush wear is accelerated in space [35], and special care must be taken to ensure that BDC Motors can still be used for aerospace applications; cost however, becomes a rigorous constraint [33].

BLDC Motors, on the other hand, use permanent magnets instead of brushes,

² Daniel Raible, Cleveland State University student.

which significantly increase a motor's lifespan but also elevate the complexity of the drive electronics. Although this is a drawback, BDLC Motors make an attractive solution because they are low maintenance motors, and in addition they provide high torques at high speeds [33]. Another motor that does not use brushes is the Stepper Motor. Even though it has the advantage of not using brushes for commutation, it has limited speed and its torque decreases with an increase in speed [34]. The advantages and disadvantages of each motor are summarized in Table I. Therefore, Brushless DC Motors were the appropriate actuator choice for the momentum wheel assembly.

TABLE I: MOTOR TYPE SELECTION: ADVANTAGES AND DISADVANTAGES [33], [34]

Brush DC Motor	Brushless DC Motor	Stepper Motor	AC Induction Motor
<p>Disadvantages:</p> <ol style="list-style-type: none"> 1. Brush to commutator arcing and wear 2. Short motor life 3. Brush dust 4. Electromagnetic interference 5. Mechanical noise 6. Low efficiency 7. Limited speed 	<p>Disadvantages:</p> <ol style="list-style-type: none"> 1. Higher electronic cost 2. Greater motor drive complexity 	<p>Disadvantages:</p> <ol style="list-style-type: none"> 1. High continuous power dissipation 2. High ripple torque 3. Limited Speed 4. Torque decrease with speed 	
<p>Advantages:</p> <ol style="list-style-type: none"> 1. Low Cost 2. Simplicity 3. Availability 	<p>Advantages:</p> <ol style="list-style-type: none"> 1. High speed (up to 100,000 RPM) 2. High torque at high speed 3. Almost double torque vs. same size BDC motor 4. Improved heat dissipation 5. No brushes – longer life time 6. Higher efficiency 7. Vacuum compatible 	<p>Advantages:</p> <ol style="list-style-type: none"> 1. Simplicity 2. Compatibility with digital control schemes 	

Once the motor type was selected, it then had to be properly sized for speed, torque and power requirements. Although, motor sizing is dependent on the type of application that the motor is used for, in general there are three parameters that govern the motor selection process: the peak torque, the root-mean-square (RMS) torque requirements, and the speed range of operation. Since this is a *high acceleration rate* application, where the motor must accelerate the load to the desired speed within a predefined time frame, the RMS torque is usually less than the peak torque. Hence, the motor torque requirement was dependent, in major part, on the peak motor torque.

The peak torque, which is the maximum torque required for the specific application, can be calculated using (3.3): the sum of the load torque T_L , the torque due to acceleration of inertia T_J , and the friction torque T_F [32], [36]. A 20% safety margin must be added in order to accommodate for additional torque requirement factors that are too complicated to be accounted for [32].

$$T_P = (T_L + T_J + T_F) \times 1.2 \quad (3.3)$$

Since the wheel inertia is the actual load in this case, T_L was combined with T_J

$$T_{Load\ Inertia} = T_L + T_J = (J_{wheel} + J_{rotor}) \cdot \alpha_{wheel} \quad (3.4)$$

where J_{wheel} is known, $J_{rotor} = 9.25 \times 10^{-5} \text{ kg} \cdot \text{m}^2$ (see Appendix A for the motor datasheet), and α is the desired wheel acceleration rate given by (3.2). Thus, (3.3) becomes:

$$T_P = \left[(J_{wheel} + J_{rotor}) \cdot \frac{\omega_{final} - \omega_{initial}}{t_{acc}} + T_{friction} \right] \times 1.2 \quad (3.5)$$

where the friction torque is given by (3.6) and is dependent on the rolling friction coefficient, which took into account an estimated safety factor of 100 ($\mu = 0.01$), and the friction caused by the rotating wheel and air.

$$T_{friction} = T_{rolling\ friction} + T_{air\ friction} \quad (3.6)$$

According to (3.6), a motor capable of producing a rotational torque equivalent to $T_p = 32.101\text{mN} \cdot \text{m}$ was needed. However, due to the restricted electric power available onboard VikSat1, additional calculations were necessary in order to ensure that the motor power rating was adequate, and that the energy required by the motor for attitude control would not exceed the power budget allocated to the ADCS (approximately 53 Watts). Figure 6 illustrates the concept of these calculations, where the power required by one motor for attitude control is equivalent to the power delivered to move the load combined with the power loss dissipated within the motor.

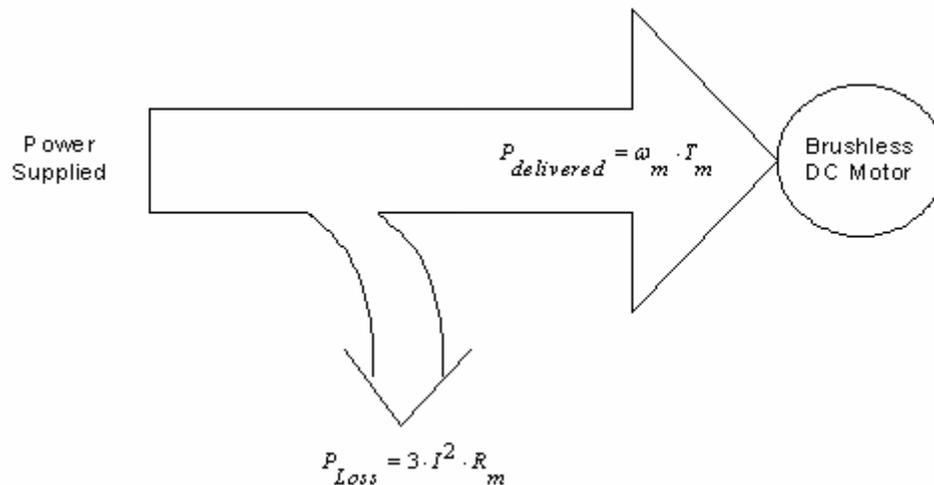


Figure 6: Electric Power Distribution – Graphical Illustration

It follows that the electric energy required to drive the load is a function of angular velocity:

$$P_{delivered}(\omega_m) = \omega_m \cdot T_P \quad (3.7)$$

where ω_m is the motor speed. As shown in Figure 6, another source of power dissipation is the motor itself. Equation (3.8) describes this type of energy loss, which is dependent on the winding resistance R_p , and the motor current I_m (the current required to drive the load to its desired speed, that is also proportional to T_P).

$$P_{loss}(I_m) = 3 \cdot I_m^2 \cdot R_p \quad (3.8)$$

Therefore, the total electric power required to drive the motor is given by (3.9); a graphical illustration is shown in Figure 7.

$$P_{Supplied}(\omega_m) = P_{loss} + P_{delivered} \quad (3.9)$$

According to Figure 7, less than 10 Watts is required to accelerate one motor and its load to 2000 RPM in one minute, which falls within the maximum allocated ADCS electric power budget of 53 Watts. Although calculations showed that 10 Watts would provide enough torque for attitude control, a 30-Watt motor was chosen as a safety margin (see Appendix A for the motor datasheet [37]).

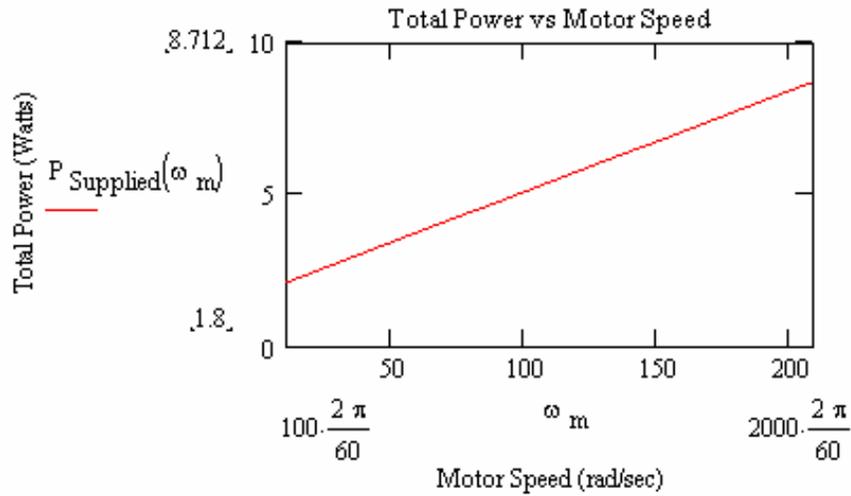


Figure 7: Total Power Delivered to the Motor

The third motor sizing requirement (motor speed range) was satisfied since, according to the motor's datasheet, this motor could operate at low speeds also. Many brushless DC motors are manufactured for high speed applications, but for such motors accurate low speed control becomes difficult. One must ensure that a motor's speed specification matches the actual operating speed range, as was the case with the momentum wheel motor.

3.2.1 Drive Electronics

Following motor sizing, the next step was to select the drive electronics required to drive the BLDC motor. The design is based on a hybrid speed controller which can, if desired, incorporate its own analog speed control by making use of the motor's Hall Effect sensors to generate its velocity feedback signal. However, the drive was set to operate in the open loop configuration since an external controller was used instead. The drive electronics board was designed specifically to accommodate the drive electronics

for both motors. This PCB is shown in Figure 8, and it was placed underneath the momentum wheel assembly (see Appendix D for the design schematics).

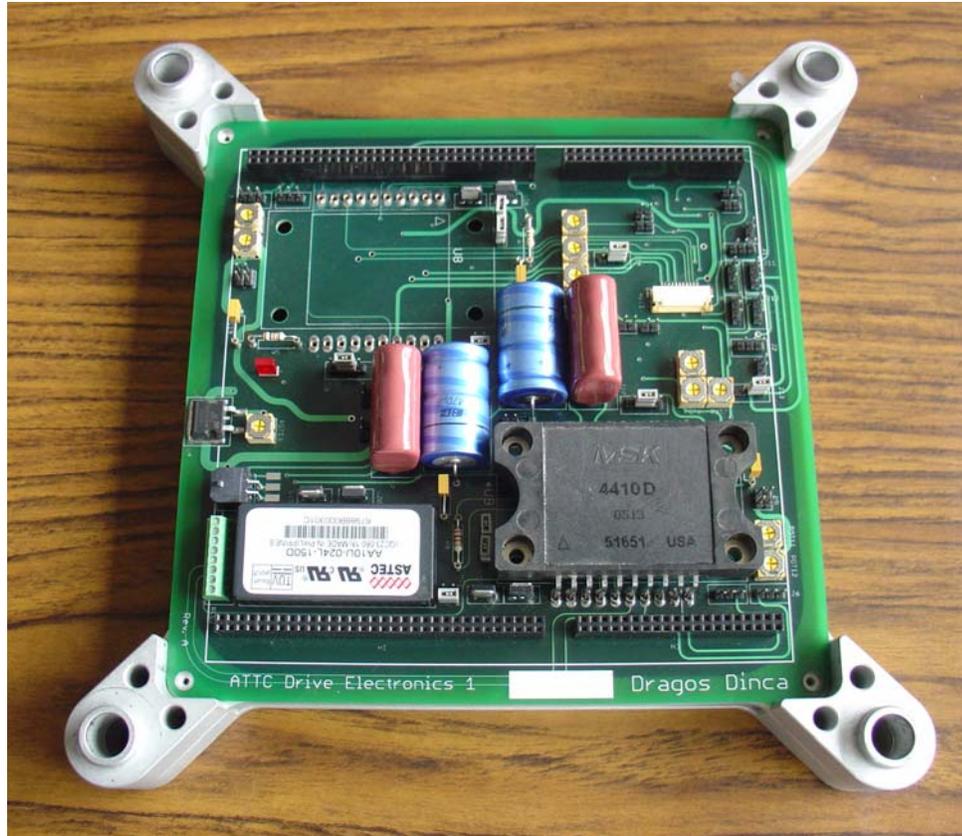


Figure 8: Drive Electronics PCB

3.2.2 Velocity Feedback Sensor

In a motion control application, feedback sensors represent a critical system component. Velocity measurement is usually accomplished by implementing either a tachometer or an encoder. The tachometer can be seen as the inverse of a motor, a DC generator, which produces an output voltage proportional to the motor shaft speed [38], [39]. The drawback with using a tachometer is its calibration accuracy and limited

resolution, as its output voltage is usually corrupted with noise. Due to the advance of electronic IC technology, encoders can also be used as velocity measurement devices by means of signal pulse differentiation. The ADCS took full advantage of this capability as two quadrature encoders were selected to measure the speed of each momentum wheel.

In contrast to the tachometer, resolution is the main advantage with using an encoder as means of providing angular velocity measurement. Depending on the type of application, a motion control system's overall performance is limited by the resolution of the measured velocity [41]. Usually, encoders are used as position feedback devices where motion is translated into a pulse signal, which is then processed by the control system [40]. Differentiating the encoder signal yields velocity measurement instead of position. However, as mentioned in Section 1.2 this can be a drawback because it reduces the processor's bandwidth.

An encoder's resolution is measured in cycles-per-revolution (CPR), where a complete revolution is equivalent to 360 mechanical degrees [40]. During the ADCS design, a 500 line encoder was initially chosen. Test results showed, however, that speed control at low velocities was not possible because the control update rate was much lower than the digital control requirement. In other words, the closed loop bandwidth was too small for controlling speeds under 200 RPM. In order to control the momentum wheels at such low RPM, a 2500 CPR encoder was used instead. Figure 9 illustrates the encoder attached to the motor shaft. One other solution that can be applied to low CPR encoders, are the specially designed encoder pulse converter ICs, which provide an output signal that can be x1, x2 or x4 the input encoder quadrature signals. Although these ICs were included in the ADCS design, they were not actually used.



Figure 9: Feedback Encoder

Another problem that occurred with the first encoder selected was its varying pulse duty cycle, which yielded false velocity measurements. The differential encoder has two output channels denoted as A and B. Although, the datasheet specified that the symmetry error between the two channels was a few electrical degrees, the error was observed to be much greater during the actual tests. All encoders do have some symmetry error and one must pay close attention to this detail, as not much can be done with a varying duty cycle but select another encoder; that of course is application dependent.

The encoders were mounted at the base of the motors and coaxial cables were used to connect them to the processing board. In order to minimize cross talk through the cable length and noise corruption due to the mechanical rotation of the momentum wheels, differential type encoders opposed to single ended were preferred. On the

processing board, differential line receiver ICs were then used to convert the differential type signals into Transistor-Transistor Logic (TTL) single-ended pulse signals.

3.2.3 Onboard Processor

A microcontroller with improved Digital Signal Processor (DSP) functionality was chosen as the onboard controller for the ADCS. The selection process was not an easy task as it was highly influenced by the complexity of the ADCS design. In addition to controlling the momentum wheels, the ADCS included the position control of VikSat1's solar panels for sun tracking with respect to the pitch axis (see Appendix B for the control electronics PCB). This was also a motion control application that included a BLDC motor, feedback sensors (solar panel position due to the sun light intensity) and drive electronics. In order to avoid additional system complexity, it was decided that one microcontroller instead of two would be used for controlling both: the momentum wheels and the solar panels. In addition to the control aspect, the onboard processor also had to communicate with other board ICs, and with VikSat1's Command and Data Handling (C&DH). Thus, in order to accommodate all these functionalities, a high pin count microcontroller was necessary and Microchip's dsPIC30F6010, a 16-bit microcontroller, was selected³. A block diagram of the ADCS functionality and interface is shown in Figure 10.

³ Andrew Jalics, Cleveland State University student; his programming insight helped with this selection early in the design process.

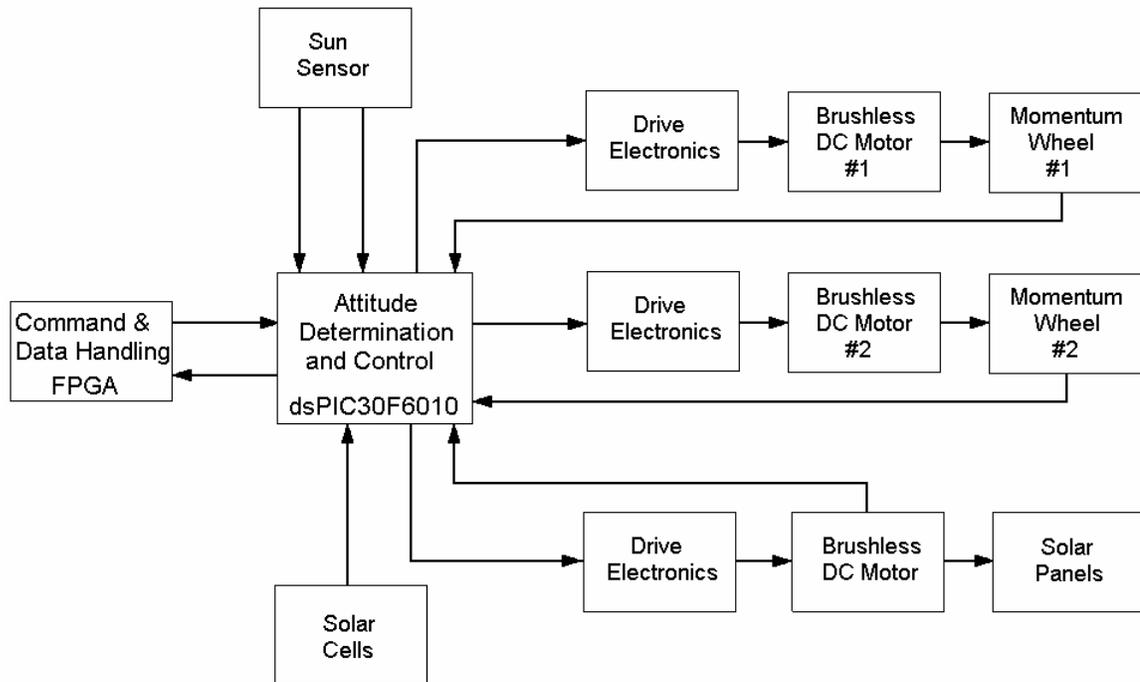


Figure 10: ADCS – Block Diagram

Since the drive electronics IC requires an analog voltage for its speed command, the control signal generated by the microcontroller was converted to an analog voltage by incorporating a 16-bit parallel port DAC. Parallel port DACs are recommended for applications that are time critical because they reduce a microcontroller's overhead programming which, in this case, helped improve the processor's bandwidth. The disadvantage, however, is that these DACs use many microcontroller input-output pins (I/Os). The trade off varies from system to system, but in the case of the ADCS, parallel DACs were chosen due to the limited microcontroller bandwidth created by its interface with the velocity feedback device. In addition, precision voltage reference ICs were used to ensure precise and stable DAC operation (these ICs also acted as buffers).

Buffer ICs were used throughout the ADCS design in order to protect the microcontroller from excessive current draws. As an example, such buffer ICs created an isolation zone between the dsPIC30F6010 and the ADCS safety switches, since these circuits were designed as safety conditions in case the motors required immediate shutdown. In another example, buffer ICs also enabled Serial Peripheral Interface (SPI) communication between the microcontroller and VikSat1's C&DH. The C&DH utilized a Field Programmable Gate Array (FPGA) IC which uses 3.3-volt logic for communication. Since the microcontroller is a 5-volt logic device, level transceiver ICs were used to accommodate for this discrepancy.

Furthermore, the microcontroller provided the necessary interface with the sun sensor used for VikSat1's attitude determination. The sun sensor, a product of Hamamatsu Photonics Co., uses a Position Sensitive Detector (PSD) board along with other electronics to produce two analog voltage signals, which represent the position of the sun in the X and Y directions [43]. A 13-bit ADC was used to process these signals and since it required a serial interface, the second dsPIC30F6010's SPI module proved very useful. In order to minimize noise corruption (glitches and bad data) with both types of converters (ADC and DAC), two separate voltage regulators were used on the actual PCB and their respective ground planes were separated.

The entire design was a complex process because it required good analog and digital circuit design knowledge. In addition, software development (microcontroller programming) was an integral part of the overall design. Prior to making the actual satellite bay PCBs, sections of the entire design were individually tested. Since the entire design was not fully integrated until the actual boards were made, the final PCB

incorporated jumpers in order to isolate different ICs, certain board areas and other circuit components. The ADCS main board (the microcontroller PCB) is shown in Figure 11 and the design schematics are attached to Appendix E.

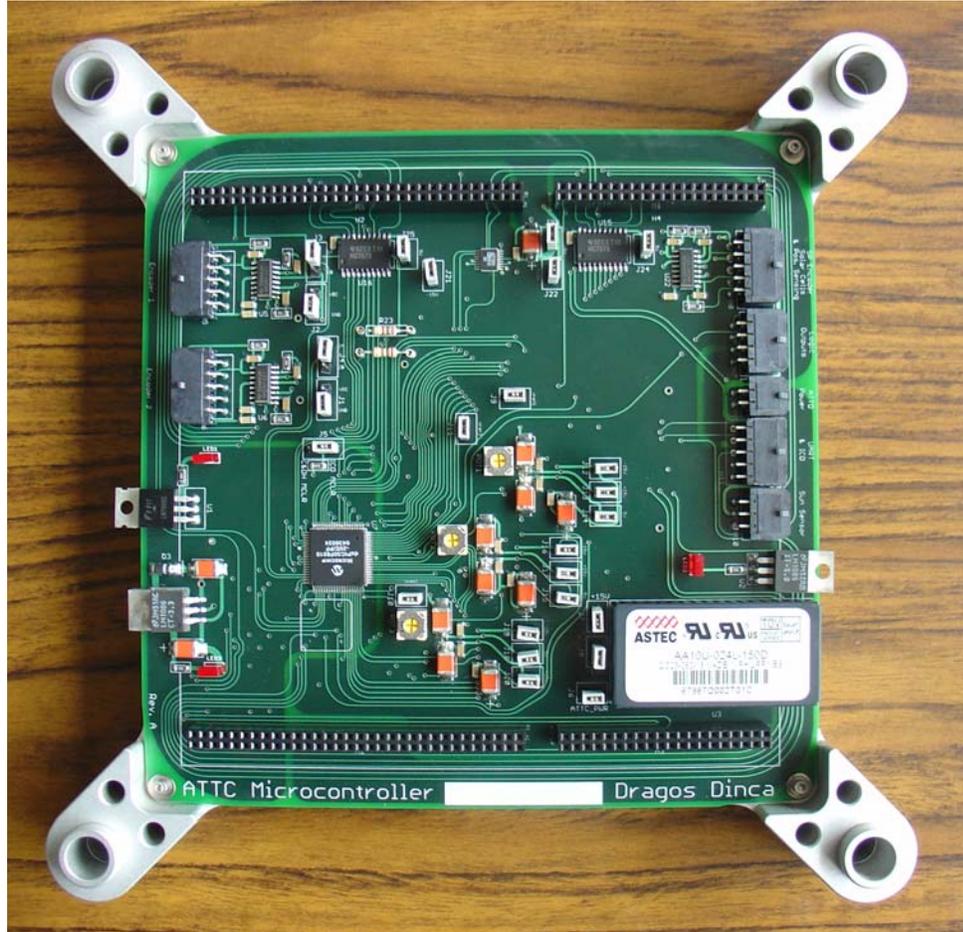


Figure 11: ADCS Main Board

CHAPTER IV

CONTROLLER DESIGN

In an attempt to overcome the friction problem and improve the performance of the ADCS, a total of three controllers were developed. The Incremental Control structure was implemented first, and it is detailed in Section 4.2. The Incremental Controller was modified, using nonlinear gains, in order to improve its ability to deal with stiction, and also its overall disturbance rejection capability (see Section 4.4). The control structure used to reformulate the friction problem in the disturbance rejection paradigm, and that can be generalized and thus applied to other systems with different types of friction is presented in Section 4.3. In order to better understand the system and for controller development purposes, the model of the momentum wheel system is outlined in Section 4.1.

4.1 The Controlled System

The first step in the controller design process was to model the momentum wheel system. The model consists of the following components: the drive electronics, the BLDC motor and the mechanical load (the momentum wheel). Since this is a linearized model, one of the assumptions made was to consider the flexible coupler between the load and the motor as a rigid connection.

The first model component, the drive electronics, was modeled as a gain block that is given by

$$K_{MSK} = \frac{V_a(t)}{V_u(t)} \quad (4.1)$$

where $V_a(t)$ is the motor armature voltage and $V_u(t)$ is the control voltage applied at the drive electronics' input (see Figure 12). In an attempt to make the model realistic, some of the drive nonlinearities were also included since the speed command input can only range from 1.5V to 4.5V. Thus, a dead zone block corresponding to the input signal's lower limit (0V to 1.5V) and a saturation block with an upper limit of 4.5V were included in the model. Since the supply voltage to the motor is 12V, a saturation limit was also added at the drive's output.

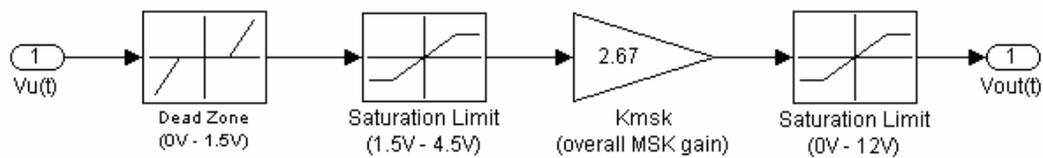


Figure 12: Drive Electronics – Block Diagram

Next, the equations used to model the BLDC motor are [44], [45]:

$$\begin{aligned} V_{az}(t) &= V_a(t) - V_{emf}(t) \\ V_L(t) &= V_{az}(t) - R_a \cdot i_a(t) \end{aligned} \quad (4.2)$$

and

$$\begin{aligned} V_{emf}(t) &= K_{emf} \cdot \frac{dy(t)}{dt} \\ V_L(t) &= L_a \frac{di_a(t)}{dt} \end{aligned} \quad (4.3)$$

where

$V_{az}(t)$ - Motor Armature Impedance Voltage

$V_L(t)$ - Motor Inductance Voltage

$V_{emf}(t)$ - Motor Back Electromotive Force (EMF) Voltage

$i_a(t)$ - Motor Armature Current

R_a - Motor Armature Resistance

L_a - Motor Armature Inductance

K_e - Back EMF Constant

Based on (4.2) and (4.3), the block diagram representation of the BLDC motor was constructed as illustrated in Figure 13.

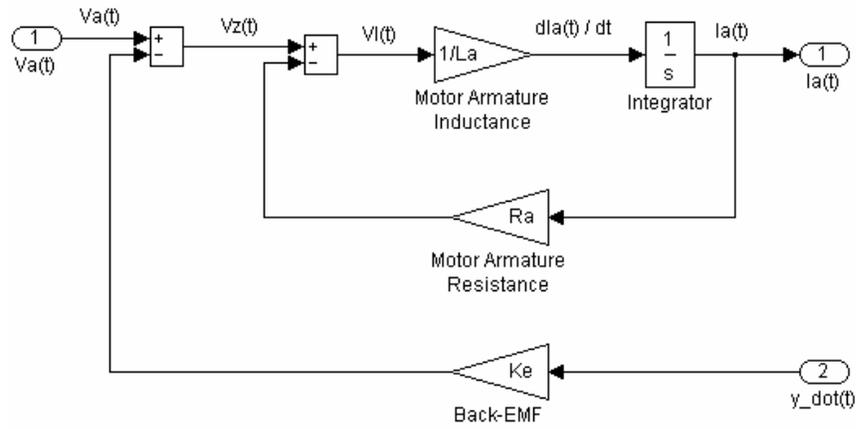


Figure 13: Motor Model – Block Diagram

The third model component, which opposes the motor motion, is composed of the actual load and the friction coefficient:

$$\begin{aligned}
 T_m(t) &= K_t \cdot i_a(t) \\
 T_{Load}(t) &= J_T \cdot \frac{d\omega_m(t)}{dt} + B \cdot \omega_m(t)
 \end{aligned}
 \tag{4.4}$$

where

$T_m(t)$ - The Motor Torque

$T_{Load}(t)$ - The Load Torque

$\omega_m(t)$ - Motor Angular Velocity Speed

K_T - Motor Torque Constant

J_T - Motor and Load Inertia

B - Friction Coefficient

Using (4.4), the mechanical model was constructed as shown in Figure 14.

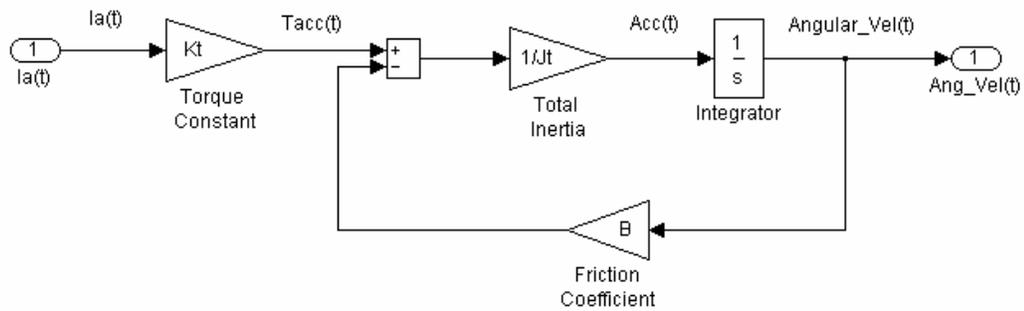


Figure 14: Mechanical Aspect of the Model – Block Diagram

The complete momentum wheel model is depicted in Figure 15. It must be mentioned however, that the motor's amperage rating of ≈ 2.4 Amps limits the amount of motor torque, $T_m(t)$, available for this application. Hence, a saturation limit with an upper limit of 2.4 Amps was added in order to mimic the real motor. Going back to the actual hardware, the drive electronics make use of an external resistor in order to limit the motor current surge.

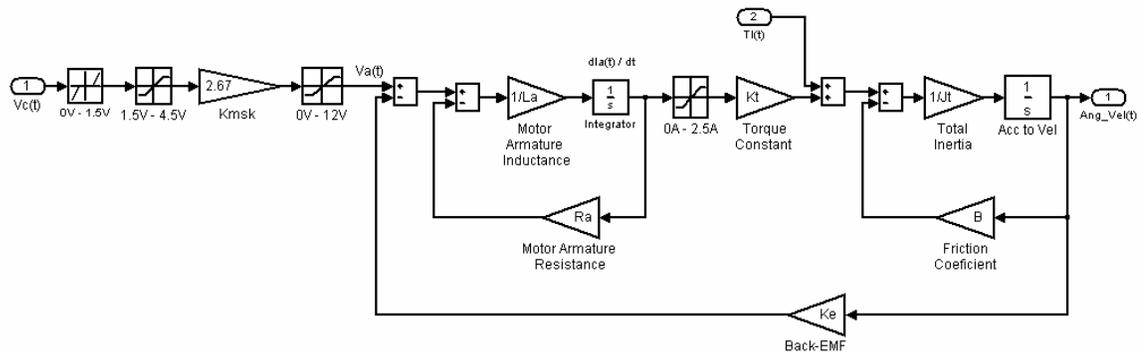


Figure 15: Complete Momentum Wheel Model – Block Diagram

Based on Figure 15, the system's overall transfer function was then obtained for analysis purposes:

$$\frac{\omega_m(s)}{V_u(s)} = \frac{\frac{K_T K_{MSK}}{R_a B}}{\tau_e \tau_m \cdot s^2 + (\tau_e + \tau_m)s + \frac{K_T K_e + R_a B}{R_a B}} \quad (4.5)$$

where τ_e is the motor's electric-time constant given by the ratio of $\frac{L_a}{R_a}$, and τ_m , the

system's mechanical time constant, is simply $\frac{J_T}{B}$.

4.2 Active Disturbance Rejection Control

The ADRC paradigm was developed as a result of trying to address the dilemma posed by modern control theory, where the controller design process is highly dependent on the mathematical model of the plant. Since mathematical models are only approximations of actual physical systems, Han questioned the robustness of a controlled system that is designed based on this methodology [51], [26]. His solution to this paradox was a control structure that can approximate the plant dynamics, based on the form of the problem, in real time [54]. Originally, this control structure was proposed under a nonlinear form and although it yielded promising results, its many gains made the tuning process quite a challenge [46], [25], [48]. However, Gao linearized it and by utilizing parameterized gains the Linear ADRC (LADRC) became much easier to tune and implement [49].

ADRC is a robust control structure because it is based on the concept of estimating the plant dynamics and compensating them, in real time through control

action, in such a way that the actual system is reduced to one of its simplest forms of $\frac{1}{s^n}$, where n is the order of the system [52]. In order to achieve this, ADRC uses an inner loop to compensate for the system's dynamics and a front end controller that provides the appropriate command following. In addition, one of its attractive features is that it can be extended in two ways: first, it can accommodate for higher order systems, and second, it can be extended to better approximate the system dynamics (provides additional system information). ADRC can thus be applied to n^{th} order multi-input multi-output (MIMO) systems [26]. Since it has been developed, the LADRC has been successfully applied to many applications such as position (point-to-point control) and tracking. However, as to the author's knowledge LADRC has not been, as of yet, implemented and applied to speed control problems.

What sets LADRC apart from other control structures is the minimal plant information required in order to control a particular system that has the form of

$$y^{(n)} = f(t) + bu \quad (4.6)$$

where y is the output, $f(t)$ represents the system's dynamics, b is the high frequency gain, u is the control input, and n represents the dominant order of the system. Since the main idea behind ADRC is that of approximating the systems dynamics, where the ideal approximation is given by (4.7), the performance of ADRC is dependent on the dominant order of the system n , and the high frequency gain b [49], [57].

$$\hat{f}(t) = y^{(n)} - bu \quad (4.7)$$

Based on (4.5), the momentum wheel system is of 2nd order. However, according to the motor datasheet the armature inductance is very small, $L_a = 0.00056 (H)$. Therefore τ_e can be neglected, and (4.5) can be simplified to

$$\frac{\omega_m(s)}{V_u(s)} = \frac{k}{s + \omega_c} \quad (4.8)$$

where

$$\omega_c = \frac{R_a \cdot B + K_T \cdot K_e}{R_a \cdot J_T} \quad (4.9)$$

$$k = \frac{K_T \cdot K_{MSK}}{R_a \cdot J_T}$$

Consequently, based on (4.8), it can be concluded that the momentum wheel system is a dominant first order system. The derivation of the 1st Order LADRC is presented next.

4.2.1 The 1st Order LADRC

The 1st Order LADRC can be applied to first, dominant first, and in some cases to second order systems. For derivation purposes consider a general first order plant such as (4.10) where y is the output, u is the input, and w represents any external disturbances. Although the coefficients a and b are unknown, some information is known about b (the high frequency gain) from the motor's dynamics and load; see equations (4.8) and (4.9).

$$\dot{y} = -ay + w + bu \quad (4.10)$$

The ADRC paradigm is able to reduce (4.10) to a simple unity gain integral plant as in (4.11), where u_0 is the new control signal which can be easily controlled with a simple front end controller.

$$\dot{y} \approx u_0 \quad (4.11)$$

In order to achieve the ideal situation of (4.11), the key idea of ADRC is to combine the internal plant dynamics and any unknown external disturbances under one term $f(y, w, t)$, also known as the generalized disturbance [26] (most often referred to as f), and (4.10) can thus be rewritten as:

$$\dot{y} = f(y, w, t) + bu \quad (4.12)$$

In addition, as long as the value of $f(y, w, t)$ can be obtained, i.e. estimated (usually denoted as \hat{f}), a control law such as (4.13) is able to achieve (4.11) by canceling in real time the plant dynamics, additional variations to it, and any external disturbances.

$$u = \frac{-\hat{f} + u_0}{b} \quad (4.13)$$

The issue now becomes that of determining \hat{f} , since the performance of ADRC is directly related to the accuracy by which f is approximated in real time. In general, a common method used to extract information when it is not measurable is through the use of observers, also known as estimators or filters [55]. An observer uses the input-output data of a plant in order to obtain state measurements or other information that is otherwise not available [55], [56]. Observers are widely used in feedback control, often times as sensor replacements since they provide vital information about state variables

that are not measurable. Therefore, in the case of this thesis an observer can be used to estimate unknown disturbances such as frictional nonlinearities.

Over time, many types of observers have been developed and a complete survey classifying them from different perspectives is presented in [55]. In this survey, the distinction is made that observer design paradigms can be classified into two categories. The first category comprises of state estimating techniques that are highly dependent on mathematical models, where the model's accuracy is directly related to the observer's performance. The other observer design method is primarily focused on disturbance estimation as it makes use of the system's input-output data to estimate, in addition to the system's states, any unknown disturbances. The estimator used by ADRC falls within the latter category and is denoted as the Extended State Observer (ESO) [55], [56].

The ESO proposed by Han, originally under the form of a nonlinear Unknown Input Observer (UIO) [56], [25] was known as the NESO or Han's observer [51], and it was designed around the key concept of including an additional state to the state space plant model in order to obtain \hat{f} . Han's Observer, however, was later linearized by Gao and it became known as the Linear ESO (LESO), which made the entire ADRC concept easier to implement and tune (the complete LESO derivation for a second order plant is detailed in [49]). In order to derive the 1st order LESO, the state space form of (4.10) can be written as

$$\begin{cases} \dot{x}_1 = x_2 + b_0 u \\ \dot{x}_2 = h \\ y = x_1 \end{cases} \quad (4.14)$$

where $x_1 = y$, $x_2 = f$ (the additional state) and $h = \dot{f}$, which is simply \ddot{y} in this case.

Based on (4.14), the state space model of (4.10) is:

$$\begin{aligned} \dot{x} &= Ax + Bu + Eh \\ y &= Cx \end{aligned} \quad (4.15)$$

where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} b_0 \\ 0 \end{bmatrix} \quad C = [1 \quad 0] \quad E = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

and b_0 is an approximation of b .

In order to estimate the states of the system, but most importantly to obtain \hat{f} , the 1st order LESO can be constructed based on the well known Luenberger observer model

$$\begin{aligned} \dot{z} &= Az + Bu + L(y - \hat{y}) \\ y &= Cz \end{aligned} \quad (4.16)$$

where L is the observer gain vector, and z represents the estimated states. Using the pole placement design technique, the LESO was parameterized by placing all observer eigenvalues at one position. Tuning became much easier as the observer gains are now a function of only one tuning parameter, ω_0 (the observer bandwidth) [49]:

$$\begin{aligned} \lambda(s) &= |sI - (A - LC)| \\ &= (s + \omega_0)^2 \\ &= s^2 + 2\omega_0 s + \omega_0^2 \\ &= s^2 + \beta_1 s + \beta_2 \end{aligned} \quad (4.17)$$

Therefore, the observer gain vector becomes:

$$L = [\beta_1 \quad \beta_2]^T \quad (4.18)$$

Optimizing the observer bandwidth, which is usually $5-10\omega_c$ (where ω_c is the desired closed loop natural frequency of the system), allows the observer states z_1 and z_2 to closely track y and $f(y, w, t)$, respectively. Once the value of $f(y, w, t)$ is known, a control law such as (4.19) is designed to reduce the plant in (4.10) to a unity gain integrator as illustrated by (4.20).

$$u = \frac{-z_2 + u_0}{b_0} \quad (4.19)$$

$$\dot{y} = (f - z_2) + u_0 \approx u_0 \quad (4.20)$$

A proportional controller can now be used to control (4.19)

$$u_0 \approx k_p(r - z_1) \quad (4.21)$$

where $k_p = \omega_c$, so that the closed-loop transfer function is pure first order [49]:

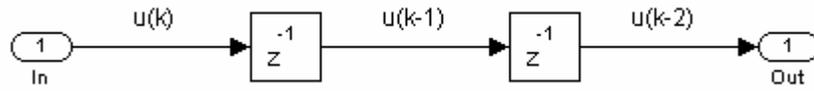
$$G_{cl} = \frac{\omega_c}{s + \omega_c} \quad (4.22)$$

Thus, the entire 1st Order LADRC is a function of two tuning parameters ω_c , and b_0 ; ω_0 may also be tuned independently if desired.

The Backward Euler discretization method (4.23) was then used to discretize the controller equations in order to implement them in the microcontroller:

$$G_c(z) = G_c(s) \Big|_{s=\frac{z-1}{T \cdot z}} \quad (4.23)$$

where z is defined as illustrated in Figure 16.

Figure 16: Representation of z

Therefore, the discretized controller equations are:

$$\begin{aligned}
 u[k] &= \frac{\omega_c(r - y) - z_2[k]}{b_0} \\
 z_1[k] &= z_1[k] + T \cdot \{ \beta_1 (y - z_1[k]) + z_2[k] + b_0 u[k] \} \\
 z_2[k] &= z_2[k] + T \cdot \{ \beta_2 (y - z_1[k]) \}
 \end{aligned} \tag{4.24}$$

In addition, due to the reduced microcontroller bandwidth, an effort was made to reduce the number of computations within the actual control loop. Thus, any redundant calculations that involved the control rate T , were parameterized and used outside the control loop:

$$\begin{aligned}
 \bar{\beta}_1 &= T \cdot \beta_1 \\
 \bar{\beta}_2 &= T \cdot \beta_2
 \end{aligned} \tag{4.25}$$

4.3 Incremental Control

Prior to LADRC, Incremental Control was the first control structure implemented and used to regulate the angular velocity of the momentum wheels. Since a control signal greater than 1.5 Volts must be maintained in order to keep a momentum wheel rotating, this control structure was developed intuitively by implementing the idea of keeping track of the “previous control signal”. As illustrated in equation (4.26), small increments based

on the error between the desired and the actual speed, are then used to make the necessary adjustments to the control signal. These small error based increments adjusted and maintained the desired momentum wheel angular velocity.

$$u(k) = u(k-1) \pm (k_i \cdot |error|) \quad (4.26)$$

Analyzing (4.26), one can conclude that this control structure is the discrete implementation of Integral Control. Although, when first implemented, this control structure was not known to be Integral Control, according to [59] a continuous time velocity algorithm is usually implemented in a discrete form by adding correction increments to the sampled time differences of the controller output. Since the output control signal is now a representation of these correction increments, the motor performs the actual integration. Velocity algorithms that are digitally implemented in such a way become known as incremental algorithms [59]. Therefore, although developed intuitively, this speed controller turned out to be an Incremental Control structure or simply Integral Control. For this application, the two terms are equivalent so they are used interchangeably. For better illustration purposes, Figure 17 shows the Incremental Control structure in a block diagram form.

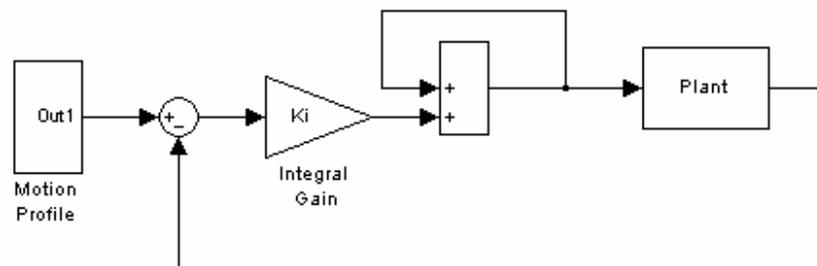


Figure 17: Incremental Control – Block Diagram Form

Integral control, in its simple form, was implemented and tested on the momentum wheel assembly. The results obtained prompted additional research. In the literature, Integral Control is often times classified as a disturbance rejection control structure or a controller that eliminates constant disturbances that are modeled as $\frac{1}{s^n}$. According to the Internal Model Principle, a controller can compensate for such disturbances by including the $\frac{1}{s^n}$ compensator in the control law [62]. Interestingly enough, ADRC also has an integrator term hidden within its control structure [54], which adds to its disturbance rejection properties. For this reason, “some form of integral control is typically included in most control systems” [58]. In addition, due to the effectiveness of Integral Control, methods have been developed in order to add integral action to State Feedback Control [58], and other control structures. Hence, if the type of disturbance to a system is known, i.e. its order n , ADRC can be improved by using another augmented state to include additional integral action to its control law, and thus more disturbance rejection. However, one must not forget that there is a trade off as integral action introduces an additional pole and thus the much unwanted phase lag, which affects the system’s stability margins.

4.4 Nonlinear Incremental Control

The no-load hardware tests with the Incremental Controller yielded promising results. However, when the controller was tested under load (with the momentum wheel)

its response to external disturbances was oscillatory in nature. This behavior was caused by the undersized motors and the integrator wind-up behavior, since the control effort available was not sufficient to quickly correct for errors. In addition, since the coefficient of static friction was fairly large, a soft start was desired in order to prevent actuator damage or slipping of the flexible coupler. Thus, an attempt was made to address these issues, by adding nonlinear gains to the Incremental Controller described in Section 4.3.

4.4.1 Nonlinear Gains

The idea of using nonlinear gains with Incremental Control resulted from the need of implementing a controller that would accommodate for the undersized momentum wheel motors. This control problem is similar to that of Time Optimal Control (TOC) where the goal is to “reach the set point in the shortest time possible with limited actuator ranges” [60]. One solution for continuous and discrete plants is the well known Bang-Bang controller, but for continuous plants this controller adds unnecessary stress to the actuators. That led to further research in this field, and a new closed-form solution for discrete plants, that is not bang-bang, was developed by Han in 1999. The new Discrete Time Optimal Control (DTOC) law was designed for a double integral plant, and the issue of control signal chattering was resolved [60], [61].

Recall that due to the encoder interface, the microcontroller had limited bandwidth. In addition, its computation time was also reduced since the microcontroller does not have true floating point units, and a C-compiler was used to emulate them instead. Therefore, combining nonlinear gains with the Incremental Control structure

was an attempt to develop a controller that would mimic DTOC (deal with the undersized motors), and at the same time its control law should have as minimum algebraic computations as possible in order to reduce the microcontroller's computation time and thus improve its bandwidth.

One alternative solution was the use of nonlinear gains. Based on the work of practicing engineers, this control method was first introduced by Han and Gao under the form known as Nonlinear PID (NPID) [46], [25], [51]. The NPID is based on the concept of using lower gains for large errors and higher gains for small errors.

Therefore combing Incremental Control with nonlinear gains and due to the nature of Incremental Control, small gains for large errors was an ideal solution in order to overcome stiction by a slow (soft) start, thus preventing integrator wind-up and overshoot. For small errors (steady state), the large gains provided disturbance rejection and minimized the steady state error. This new proposed control structure is denoted as Nonlinear Incremental Control (NIC), and its control law is:

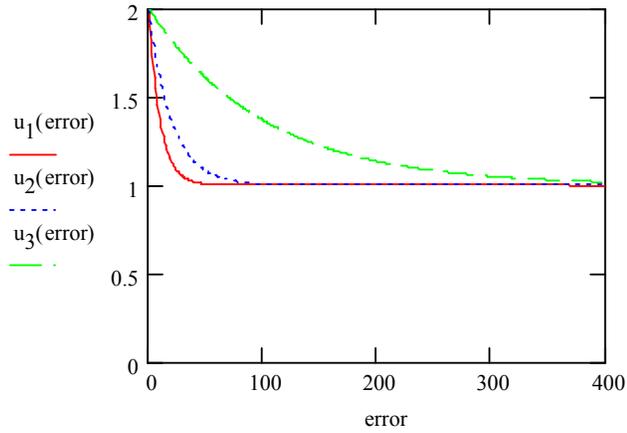
$$u(k) = u(k-1) \pm u_{\max} \cdot \left(\beta + e^{-\alpha \cdot |\text{error}|} \right) \quad (4.27)$$

where $u(k)$ is the current control signal, u_{\max} is the maximum control effort available or that can be provided by the actuator without exceeding its limit (range), α ($0 < \alpha < 1$) and β adjust the slope and the initial control effort, respectively. For illustration purposes consider Figure 18, where $u_{\max} = 1$ and $\beta = 1$, and Figure 19 where $u_{\max} = 1$ and $\beta = 0.1$.

$$\alpha_1 := 0.1 \quad u_1(\text{error}) := u_{\max}(\beta + e^{-\alpha_1 \cdot |\text{error}|})$$

$$\alpha_2 := 0.05 \quad u_2(\text{error}) := u_{\max}(\beta + e^{-\alpha_2 \cdot |\text{error}|})$$

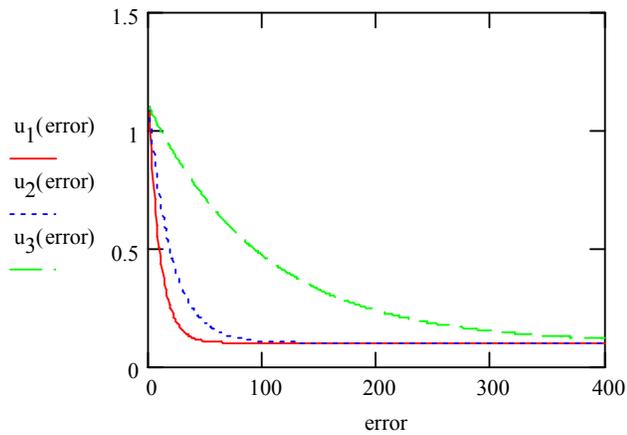
$$\alpha_3 := 0.01 \quad u_3(\text{error}) := u_{\max}(\beta + e^{-\alpha_3 \cdot |\text{error}|})$$

Figure 18: NIC – Graphical Illustration ($\beta = 1$)

$$\alpha_1 := 0.1 \quad u_1(\text{error}) := u_{\max}(\beta + e^{-\alpha_1 \cdot |\text{error}|})$$

$$\alpha_2 := 0.05 \quad u_2(\text{error}) := u_{\max}(\beta + e^{-\alpha_2 \cdot |\text{error}|})$$

$$\alpha_3 := 0.01 \quad u_3(\text{error}) := u_{\max}(\beta + e^{-\alpha_3 \cdot |\text{error}|})$$

Figure 19: NIC – Graphical Illustration ($\beta = 0.1$)

For the momentum wheel application, (4.27) was parameterized for one tuning parameter, α :

$$u(k) = u(k-1) \pm \text{nonlinear} \quad (4.28)$$

where

$$\text{nonlinear}(u_{\max}, x, \text{error}, \alpha) = \begin{cases} u_{\max} \cdot (\alpha + e^{-\alpha|\text{error}|}), & |\text{error}| > x \\ 0 & |\text{error}| < x \end{cases} \quad (4.29)$$

When the error approaches zero, in order to avoid an infinite gain and thus constant chattering around the set point, the previous value of the control signal is maintained (that limit is set by x). Test results have shown that the controller becomes more aggressive with higher β values, thus minimizing steady state error and improving command following. On the same note, the tuning parameter α can be tuned for a more aggressive soft start.

CHAPTER V

HARDWARE IMPLEMENTATION

The three controllers detailed in Chapter IV were implemented and tested on the momentum wheel system, and some of the implementation issues are addressed in Section 5.1. The actual hardware results are presented in Section 5.2, followed by the controller comparison which is outlined in Section 5.3.

5.1 Controller Implementation Issues

Different implementation issues (software, electrical and mechanical) were encountered during the controller implementation process. The most critical issues are mentioned in Section 1.2, and additional supporting information pertaining to these issues is presented next.

5.1.1 Encoder – Microcontroller Interface

An important feature of the dsPIC30F6010 is the Input Capture Module (ICM), which facilitates frequency (period) and pulse measurement. This module was utilized in order to determine the encoder pulse frequency, which was then used to calculate the momentum wheel angular velocity. Although great resolution was achieved, there were challenges associated with his method as it reduced the processor bandwidth and thus the available control rate.

The ICM uses an internal timer to capture the time period between rising and/or falling edges of the input signal (the pulsed encoder signal in this case). This process is analogous to putting a time stamp to every captured encoder period. Thus, within the microcontroller, the ICM generates an interrupt when an encoder pulse is detected, such as a rising or a falling edge. To measure a full pulse period, a second interrupt is generated when a second rising or falling edge is sensed. A free running timer is used to determine the time difference between the two interrupt events. That time frame (T_{ICM}) was then multiplied by the microcontroller's instruction cycle period ($T_{CY} = 0.2\mu s$) in order to obtain the time duration of the encoder pulse in real time (T_{real_time}):

$$T_{real_time} = T_{ICM} \times T_{CY} \quad (5.1)$$

Once the real time period of the encoder pulse was known, the angular velocity of the motor in RPM was calculated as follows:

$$Motor_RPM = \frac{1}{T_{real_time}} \times \frac{1}{2500CPR} \times \frac{60sec}{1min} \quad (5.2)$$

Since the microcontroller processes the information sequentially, the processor was paused while the encoder pulse period was measured. Therefore, the processor bandwidth and thus the available control rate became a function of the encoder pulse time duration, which decreased as the motor speed increased. That led to a varying time capture within the microcontroller and thus to a varying control rate. At low motor speeds, the pulse duration was larger than at high speeds, which resulted in a lower control rate than at higher speeds. Since a fixed control rate was desired, a second timer was utilized (within the velocity measuring routine) to achieve that. This second timer was analogous to a wait routine for the high frequency encoder pulses. In addition, depending on the momentum wheel speed, a total of three velocity routines were used in order to accommodate for the varying pulse frequency.

The velocity measuring routine (encoder interface) significantly reduced the microcontroller's bandwidth. Another factor that also contributed to the reduced control rate was the use of emulated floating point numbers. The dsPIC30F6010 does not have true floating point capability, and the C-compiler emulates the floating point numbers instead. Thus, additional time was required to compute the mathematical operations that involved variables of the floating point type. Usually this is not an issue but due to the already reduced control rate, any other delay had to be taken into account. For this reason, the computation time for each controller was measured (see Table 5.1).

In addition to the time required to compute the controller equations, the time duration for all other programming routines that played a part in the overall control rate were also measured: the interrupt routines, the velocity measurement routine (first and second capture events and additional code) and, although it was not as significant, the

code used to control the DAC in order to output the control signal. It must be mentioned that in order to avoid the generation of false output voltages, the DAC uses two registers that act as a double-buffering zone. Thus, two DAC control lines were used to latch the control signal data into each register prior to being converted to actual voltage. Therefore, the total control rate was a function of the controller computation time, and the total time taken by the previously mentioned programming routines.

5.1.2 Sensor Noise

One other constraint mentioned in Chapter I, was that of sensor noise. As it was explained in Section 1.2, the speed measurement accuracy was affected by sensor noise which was induced by the onboard electronics, and by the interconnecting mechanical components. Thus, the first type was the high frequency noise (the voltage spikes on the falling and rising edges of the encoder signal), which originated from the encoder electronics (see Figure 20). The second type of noise was due to the vibrations caused by the high speed rotational motion of the momentum wheels. Since the optical encoder was mounted on the motor shaft, the mechanical vibrations affected the encoder's optical interface. The optical encoder utilizes a light source, such as a Light Emitting Diode (LED), to shine light through a codewheel. At the other end a photodetector IC is used to generate the encoder pulses. Since the light source and the codewheel must be in sync, the mechanical vibrations altered the encoder's accuracy. It was observed that the quality of the encoder signal degraded with increased momentum wheel speed.

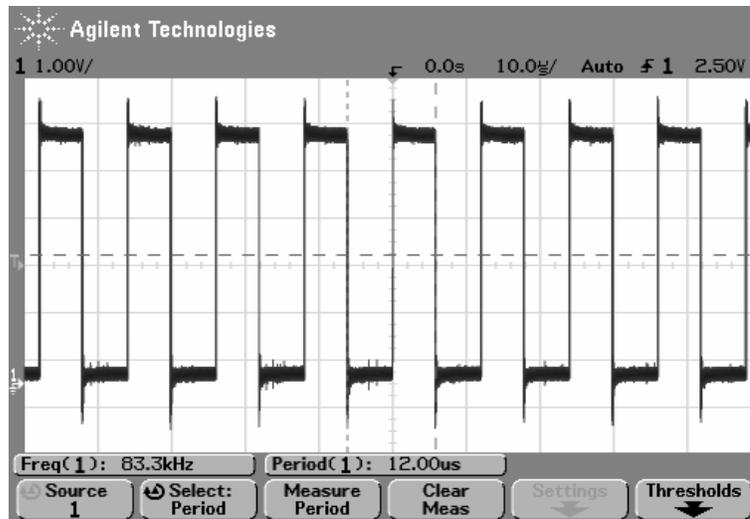


Figure 20: Encoder Signal – With Load at High Speeds

In order to minimize the overall effect of sensor noise and thus improve the quality of the encoder signal, the ADCS design included coaxial cables and driver signal ICs (as described in Section 3.4). Although this design practice prevented further signal deterioration, it could not eliminate the voltage spikes that appeared on the signal's rising and falling edges. To minimize these spikes, pull-up resistors (2.2k Ω) were added to each encoder channel but the signals did not improve much. A 22uF capacitor, however, reduced the voltage spikes and the ringing that occurred on top of the encoder pulse (see Figure 21). In addition, averaging was also used within the microcontroller to account for any false ICM triggering. These two solutions attenuated the overall measurement noise.

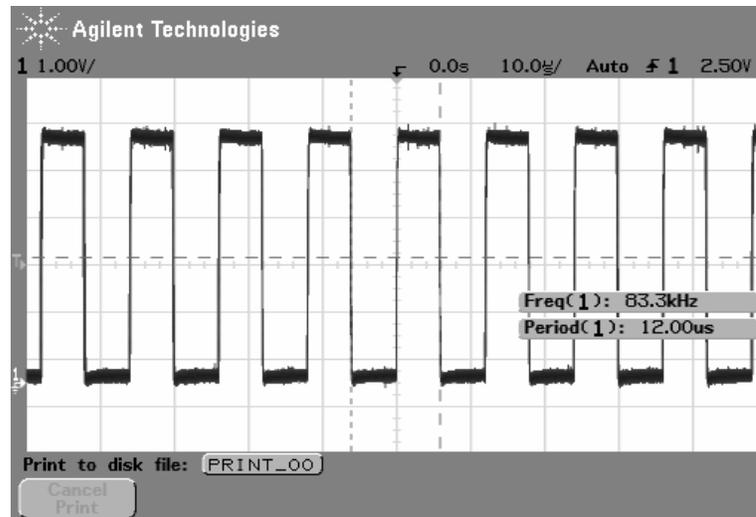


Figure 21: Improved Encoder Signal

5.1.3 Motion Profiles

Although a step input can be used for simulation purposes, it is impractical in actual hardware. Gradual set point changes, also known as motion profiles (S-curve, trapezoidal etc.), are used instead. Since the momentum wheel actuators have limited range, the use of profiles was necessary in order to keep the error small and avoid saturation. Three motion profiles have been used while testing the three controllers.

The first profile was developed based on the concept of feedback, where the measured angular velocity was fed back to the reference set-point, as exemplified in Figure 22. Hence, this motion profile was generated by incrementing the set point only when the previous set-point value was reached. The advantage consists in allowing the controller to be aggressive, while ensuring that the actuator limits are not exceeded. The profile's bandwidth can be adjusted by changing the intermediary steady state limits.

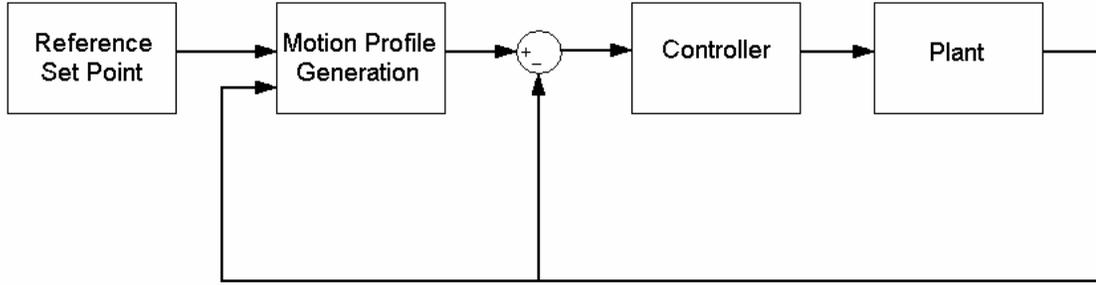


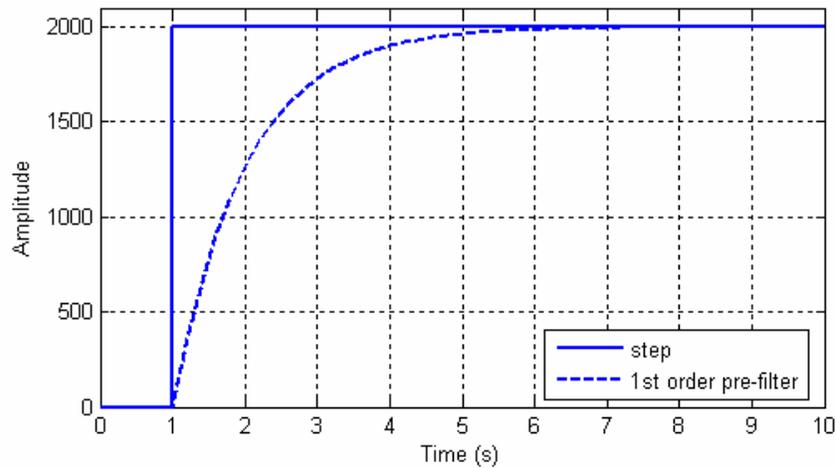
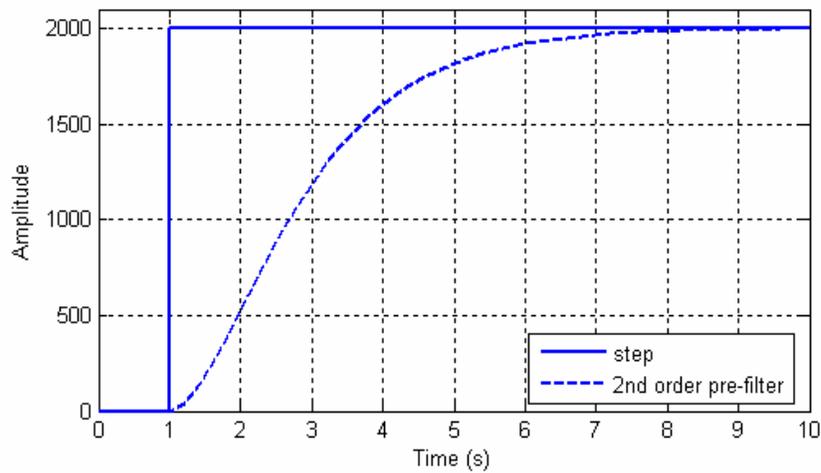
Figure 22: Feedback Based Motion Profile – Block Diagram

The feedback based motion profile was successfully implemented and tested. However, in order to fairly compare the three controllers discussed in Chapter 4, a fixed point profile was required instead. Pre-filters, such as first or second order transfer functions, are good alternatives as they are used after a step set-point change. The first and second [63] order pre-filters are given by (5.3) and (5.4) respectively, and both pre-filters can be tuned by their bandwidths ω_n :

$$\frac{R'(s)}{R(s)} = \frac{\omega_n}{s + \omega_n} \quad (5.3)$$

$$\frac{R'(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\omega_n s + \omega_n^2} \quad (5.4)$$

where $R(s)$ is the input step set point change, and $R'(s)$ is the filtered output response. For illustration purposes, the two profiles are compared to a step change with an amplitude of 2000 (see Figures 23 and 24 for the first and second order transfer functions, respectively). Notice the slope difference between the profiles. In addition, although it takes more time to reach steady state, the second order pre-filter offers a soft (smoother) motion start than the first order.

Figure 23: Pre-filter Profile – 1st OrderFigure 24: Pre-filter Profile – 2nd Order

5.1.3 Data Acquisition

The performance of the speed controllers described in Chapter IV was analyzed based on the following test data: the momentum wheel angular velocity, and the control signal (both measured in real time). To obtain this data, a second microcontroller was

used since the first dsPIC30F6010 had limited bandwidth due to the encoder interface. The second microcontroller was implemented on a separate breadboard and used for data acquisition only. Thus, from the ADCS board, parallel jumper wires were used to connect the output control voltage and the input encoder feedback signal to the additional breadboard. The control signal was recorded using the microcontroller's internal 10-bit ADC module, and the same velocity routines were programmed to measure the speed of the momentum wheel. Real time was added to each corresponding data in order to make the test data realistic. In addition, since the acquisition time varied (depending on the amount of data sent to the computer) an internal timer was utilized for a constant data acquisition time of 40 milliseconds.

Once the test data (angular velocity, control signal and real time) was acquired and temporarily stored, it was then sent to the computer and saved to a data file (*.dat*). Communication between the microcontroller and the desktop computer was realized using the microcontroller's Universal Asynchronous Receiver Transmitter (UART) module (a full-duplex asynchronous communications system), and the desktop computer's Hyper Terminal (baud rate of 9600 bits-per-second). Since the Hyper Terminal utilizes an RS-232 interface that operates at ± 10 Volts, a transceiver IC (the MAX202) was utilized in order to convert the 5-Volt microcontroller logic. In addition, to ensure reliable data acquisition, the two microcontrollers were programmed to operate in sync. Jumper wires and interrupt routines were used to facilitate communication between the two processors so that both start at the same time. In order to better understand the overall data acquisition process, a representative block diagram is shown in Figure 25.

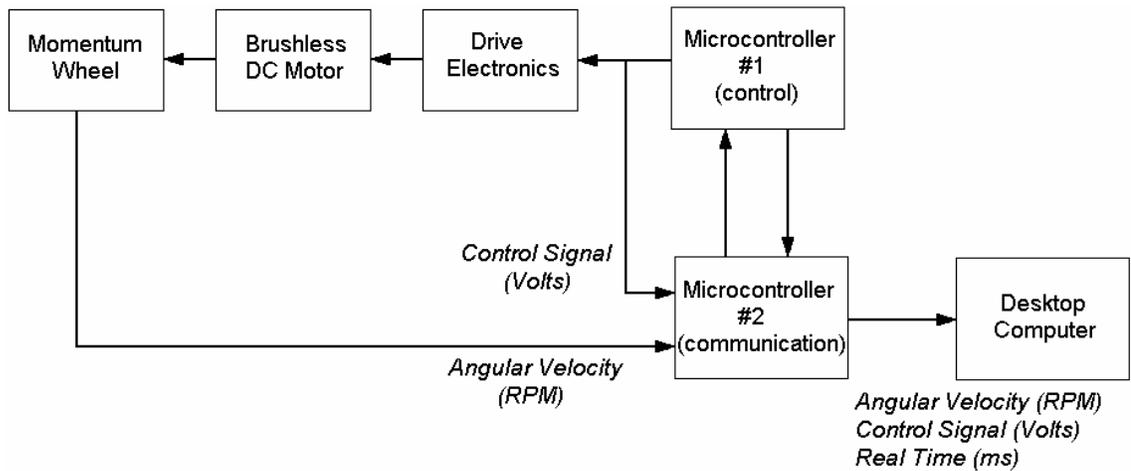


Figure 25: Data Acquisition – Block Diagram

5.2 Hardware Results

Prior to connecting the load (the momentum wheel), the speed controllers (Incremental Control, 1st Order LADRC and NIC) were initially tested under no load (only the motor itself). To test the robustness of each controller (invariance to plant parameter variations and unmodeled system dynamics such as friction), external disturbance were induced. Once the controllers were tested for their respective disturbance rejection capability, the momentum wheel was connected to the motor. In addition, as in the no load case, for further analysis purposes external disturbances were induced also.

5.2.1 No Load Hardware Results – Without External Disturbance

The purpose of these tests, *under no load and without any external disturbances*, was to examine each controller's initial response, verify their basic functionality, and familiarize with their respective tuning parameters. Although the controllers were initially tested with a step set-point change of 2000 RPM, the motion profile given by (5.3) was discretized using (4.20) and its equivalent difference equation was implemented in the microcontroller instead. The tuning of each controller is discussed next, where all controller gains are in their analog form (do not include the sample time parameterization).

The first controller implemented and tested was the Incremental Controller. Its initial tuning was quite simple as test data showed that high gains cause overshoot. For the no load case k_i was set 100, and the motor speed response is depicted in Figure 29.

The LADRC has three tuning parameters: k_p , ω_o , and b_0 . All three parameters play an important role in the overall response of the system: transient, steady-state error, disturbance rejection, control signal noise attenuation, and system stability. Tuning began by setting the high frequency gain, b_0 . Based on equation (4.8) and (4.9), b_0 was set to 5200. The other gains, ω_o and k_p , were then tuned for optimal performance. The second tuning parameter ω_o , the observer bandwidth by which the observer tracks the states, was increased to the point where the control signal noise began to deteriorate: $\omega_o = 100$. The gain k_p , (the desired closed loop natural frequency) as it was explained in Section 4.3, is usually selected based on the desired transient performance criteria, i.e. the

desired settling time. Since the purpose of the no load tests was to compare the three controllers' overall performance, there was no desired settling time but an effort was made to have a fast transient, and k_p was tuned to 10000. It must be noted that these parameters were also tested in simulation (see Figure 26 for the continuous time domain simulation set-up).

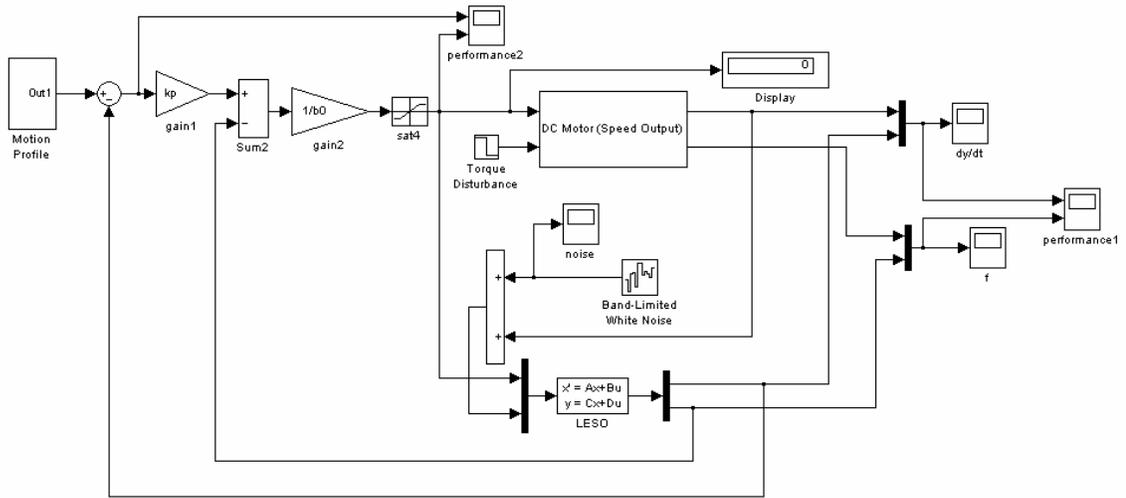


Figure 26: LADRC – Simulation Set-up

The tuning process, however, revealed two interesting observations. First, it was intriguing to notice that although the theoretical value for the high frequency gain was simulated in Simulink, a lower value for b_0 yielded better results when implemented in the actual system. The difference between the continuous time simulation and the actual hardware implementation consists of the DAC gain, $k_u = \frac{5 [Volts]}{2^{16} - 1}$, which must be accounted for. Thus, the control law given by (4.19) is rewritten as

$$u = \frac{u_0 - z_2}{b_0} \cdot k_u \quad (5.5)$$

which means that in order to account for the DAC gain, b_0 must be lower. In addition, b_0 also appears in one of the implemented discretized controller equations: $T \cdot b_0 \cdot u$, where T is the sampling time ($T = 0.0012$ seconds in this case). Notice that in the continuous domain, u ranges from 0-to-5 Volts, but within the microcontroller u is in bits (0 to 65535). Thus, for implementation purposes a lower b_0 value must be used instead.

Another observation, for this system, was the linear relationship between k_p and b_0 , which showed to improve the controller's disturbance rejection capability. Consider the control law block diagram depicted in Figure 27, which can be rearranged as shown in Figure 28. The second configuration explains the following: first, as observed during the tuning process, there is linear relationship between k_p and b_0 , and second, notice that high b_0 values do limit the effectiveness of \hat{f} . In conclusion, disturbance rejection improves with higher k_p and lower b_0 values, respectively. Thus, for the no load case the 1st Order LADRC was tuned to $k_p = 500$, $\omega_o = 100$, and $b_0 = 120$; the motor response is shown in Figure 30.

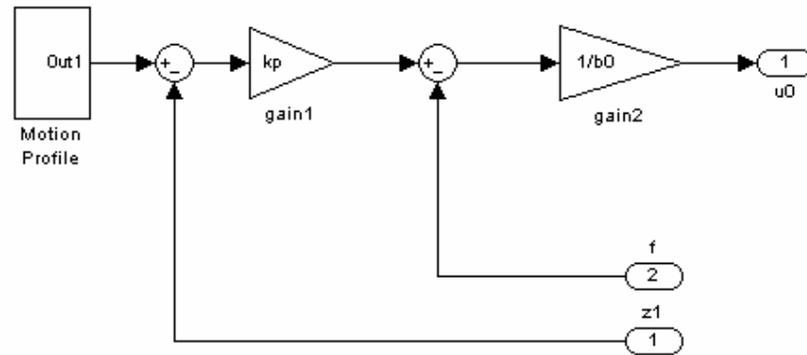


Figure 27: LADRC – Front End Control Law

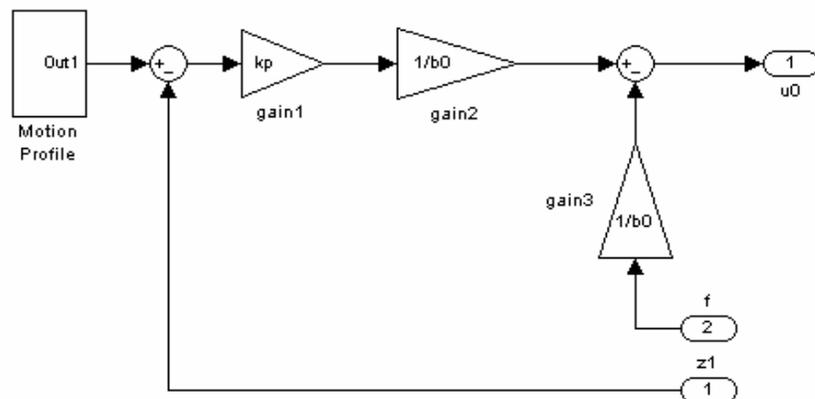


Figure 28: LADRC – Front End Control Law (Reduced Block Diagram)

The NIC tuning was simple and intuitive. First, u_{\max} was set to the maximum control effort available. The tuning parameter α , was then adjusted for a fast transient ($\alpha = 0.1$), while β was set to 0.5 for steady state tracking. Lastly, the steady state limit x , was set to 2 RPM. The tuned response for the no load case is illustrated in Figure 31.

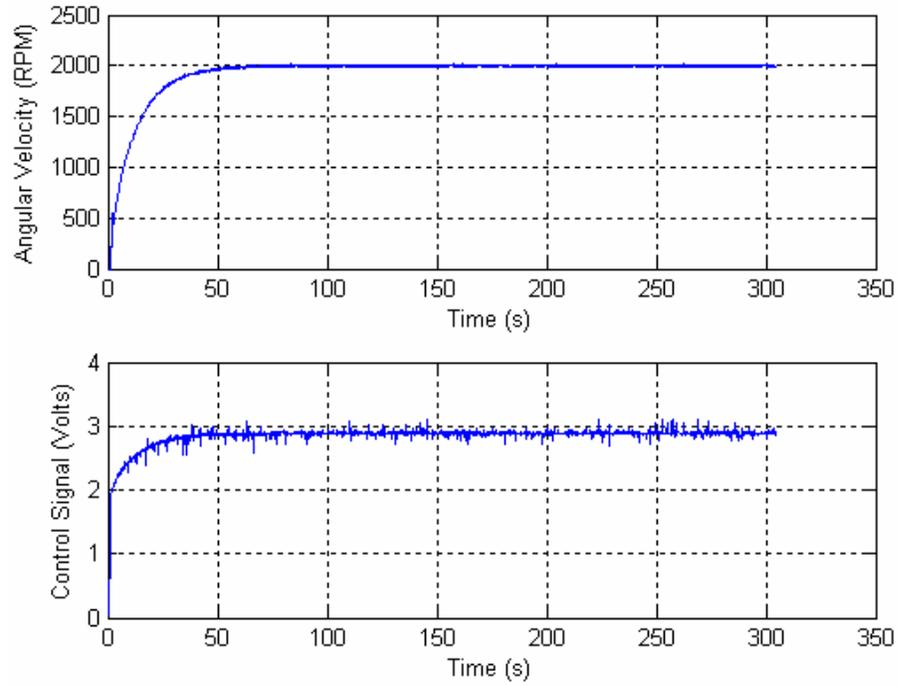


Figure 29: Incremental Controller – No Load (Without Disturbance)

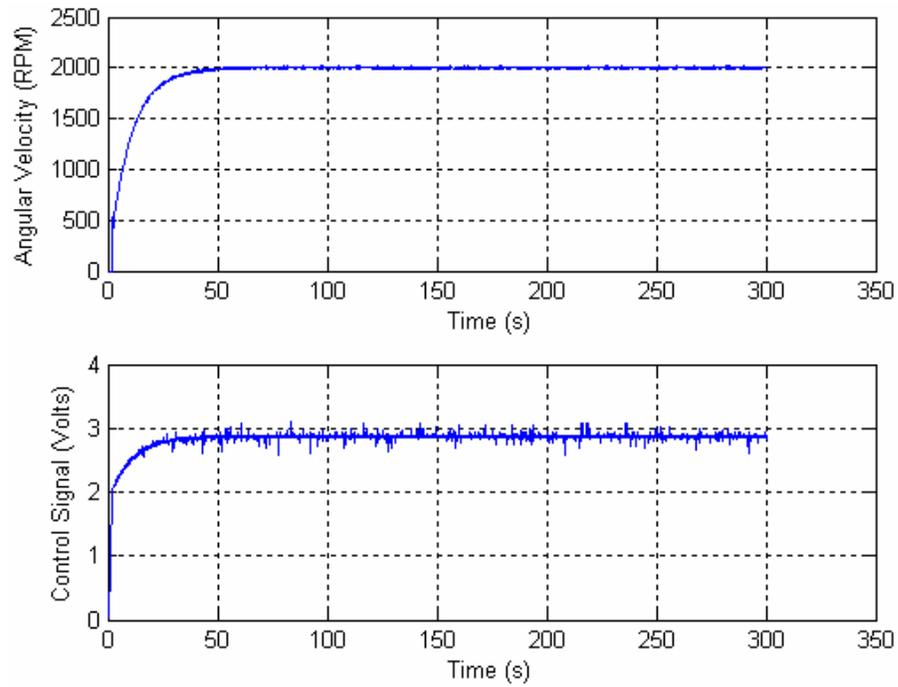


Figure 30: LADRC – No Load (Without Disturbance)

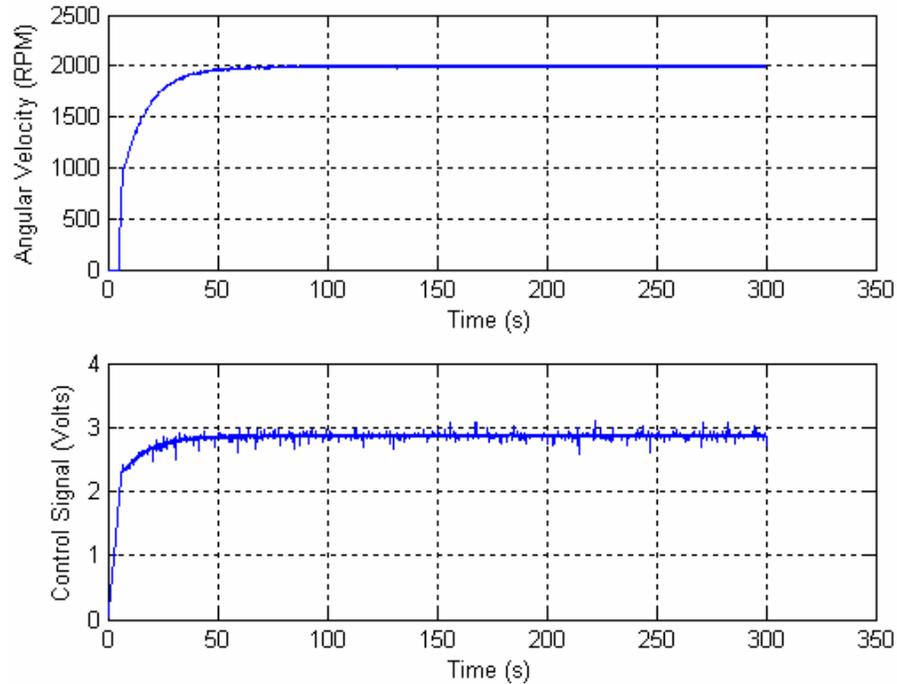


Figure 31: NIC – No Load (Without Disturbance)

5.2.2 No Load Hardware Results – With External Disturbance

In order to test the disturbance rejection capability of all three controllers under no load, different types of speed disturbances were induced. Thus, when the motor speed reached steady state, the motor was briefly stopped. Having recovered from that initial impulse, other disturbances with varying magnitudes (the motor being slowed down by different RPM) were also induced. Each controller's individual response, Incremental Control, 1st Order LADRC, and NIC is shown in Figure 32, Figure 33, and Figure 34, respectively.

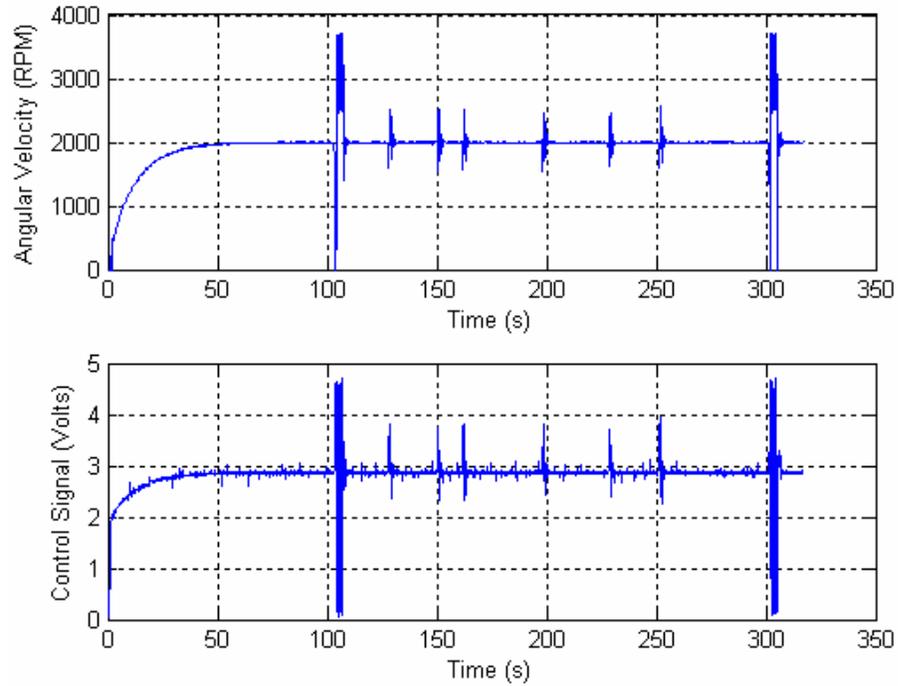


Figure 32: Incremental Controller – No Load (With Disturbance)

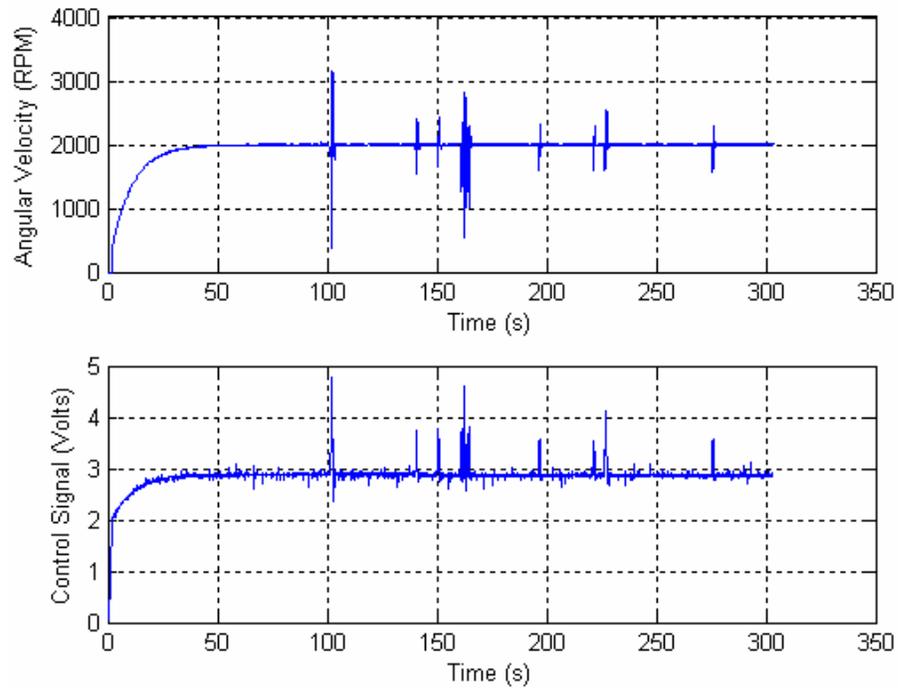


Figure 33: LADRC – No Load (With Disturbance)

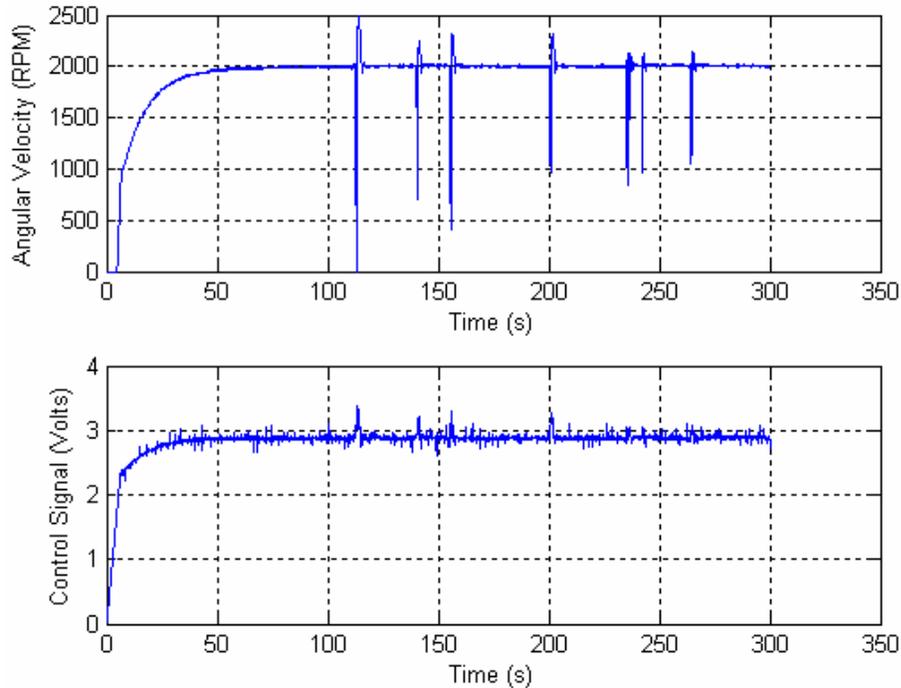


Figure 34: NIC – No Load (With Disturbance)

5.2.3 Under Load Hardware Results – Without External Disturbance

Connecting the momentum wheel to the motor required that all three controllers be retuned. In addition, due to the high static friction coefficient, the two motion profiles given by (5.3) and (5.4) were combined into one. Therefore, since a soft start was desired, the second order pre-filter was used for speeds less than 400 RPM. Once the momentum wheel velocity reached the 400 RPM mark, the first order pre-filter was then used instead. It must be noted that the combination of the two profiles was only used with the Incremental Controller and the 1st Order LADRC. The NIC made use of the first order pre-filter only, since the NIC was designed for applications with high stiction or application that require a soft start. Retuning of all three controllers is presented next.

To prevent integrator wind-up, the Incremental Controller was made less aggressive and its gain was changed to 40; the motor response is illustrated in Figure 35.

For the same reasons, the NIC was also re-tuned. Initially, u_{\max} was set to 50% of its previously used value but then tuned to 25%. Since a soft start was desired to overcome stiction and to avoid slipping of the bellows coupler, α was left unchanged. Although the second tuning parameter β can reduce the steady state error if increased, it was left unchanged because higher values showed a deterioration in the transient response. The NIC retuned response (with load) is shown in Figure 37.

The 1st Order LADRC gains were also retuned since the system's inertia changed due to the momentum wheel and the high friction coefficient. Thus, based on an estimated friction coefficient of $\mu = 0.001$ the theoretical value for b_0 was recalculated to $b_0 = 43$. However, based on experimental data the controller parameters were tuned to: $k_p = 50$, $\omega_o = 100$, and $b_0 = 2$; the system's response is illustrated in Figure 36.

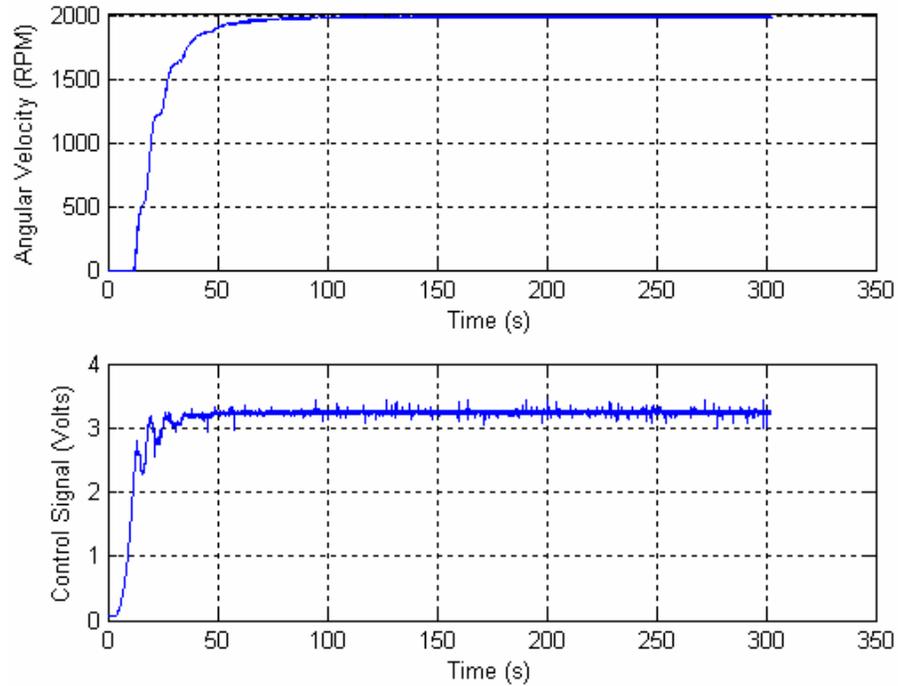


Figure 35: Incremental Controller – Under Load (Without Disturbance)

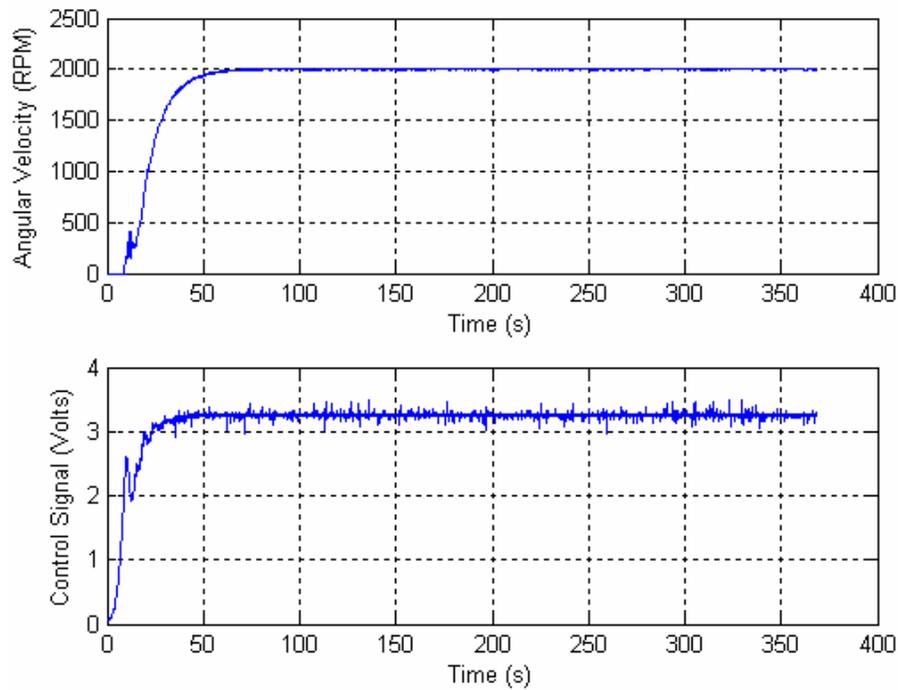


Figure 36: LADRC – Under Load (Without Disturbance)

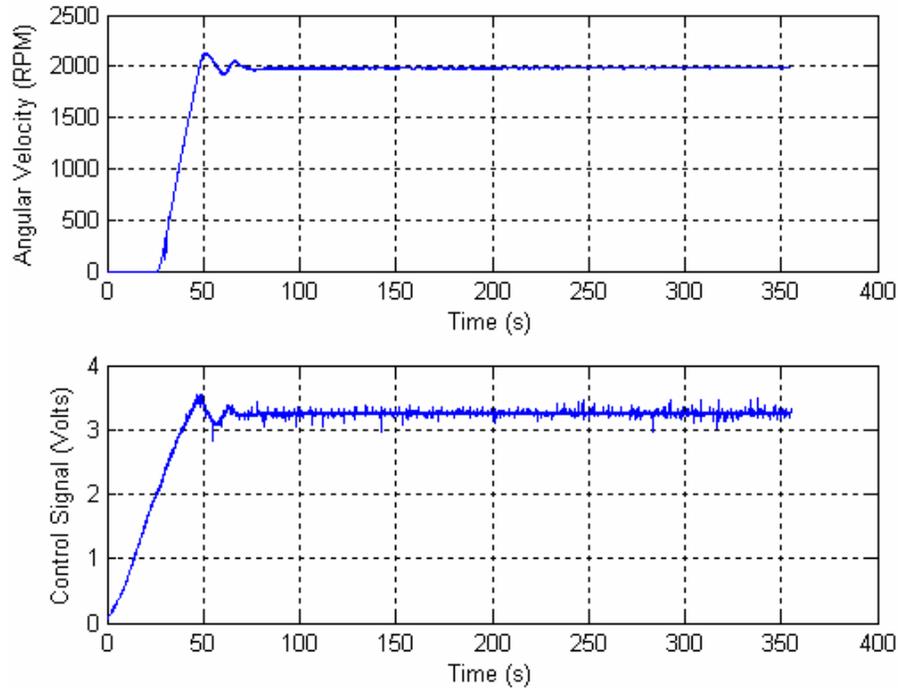


Figure 37: NIC – Under Load (Without Disturbance)

5.2.4 Under Load Hardware Results – With External Disturbance

In order to further test each controller, as in the no load case, two external disturbances were added: first, the motor speed was reduced by 200 RPM; once the system recovered (steady state) the motor was slowed down again by 150 RPM. Since, in the previous section, each controller was tuned for disturbance rejection, the tuning parameters for each controller were left unchanged. The response of each control structure, Incremental Control, 1st Order ADRC, and NIC is shown in Figure 38, Figure 39, and Figure 40, respectively.

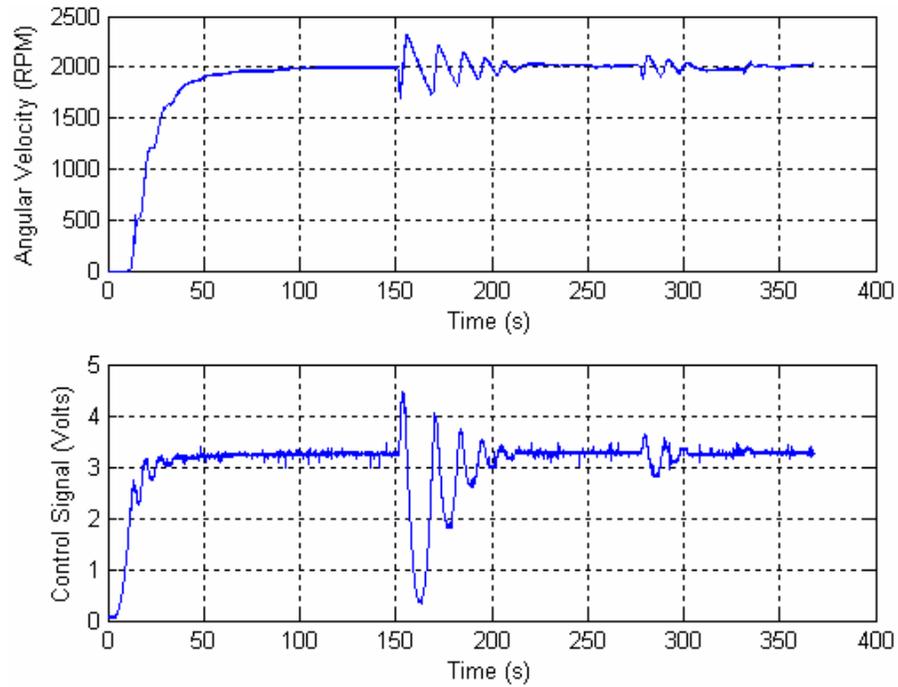


Figure 38: Incremental Controller – Under Load (With Disturbance)

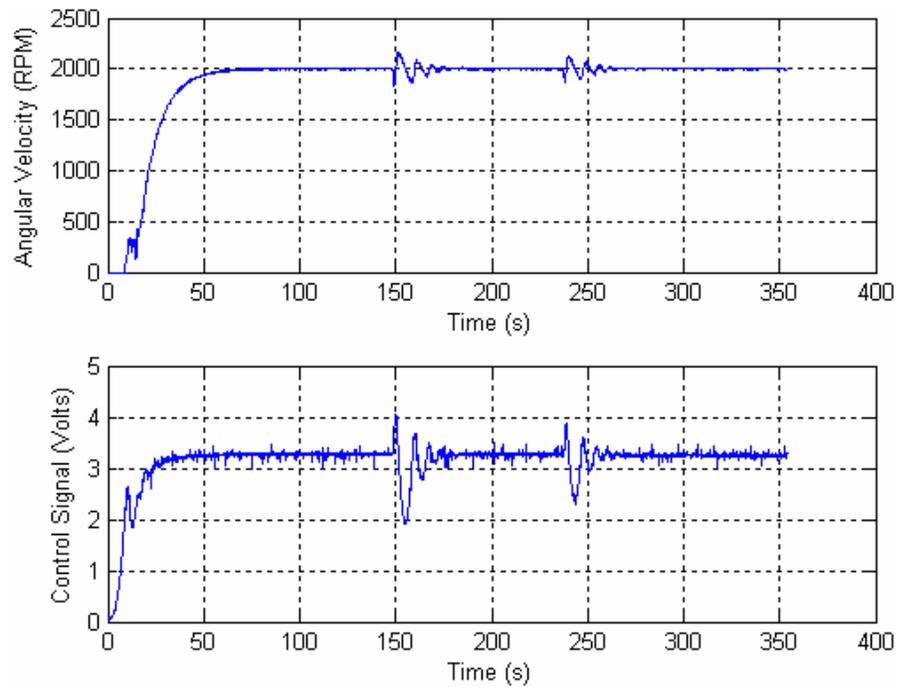


Figure 39: LADRC – Under Load (With Disturbance)

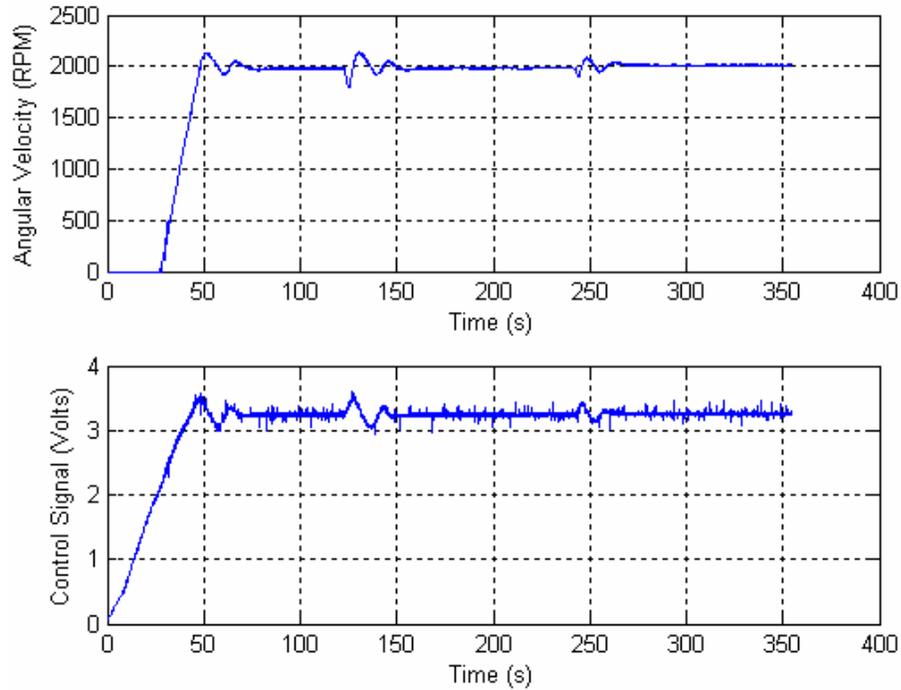


Figure 40: NIC – Under Load (With Disturbance)

5.3 Controller Comparison

For this application, due to the high friction coefficient, the comparison criteria for these controllers consist of the following [64]:

- 1) Steady state tracking: minimum steady state error.
- 2) Controller robustness: tolerance to dynamic variations, i.e. friction (including static) and inertia change.
- 3) Disturbance rejection: minimum recovery time (the external disturbances induced have approximately the same magnitude in RPM).

- 4) Controller tuning simplicity: number of tuning parameters and general ease of tuning.
- 5) Control signal monitoring (amplitude and frequency).
- 6) Controller computation time: time duration for the implemented controller difference equations.

These criteria only apply to the results detailing steady state angular velocity, since the Incremental Controller and the NIC are considered to be tracking controllers while the LADRC, in its original form, is not. Therefore, profile tracking during transient is not considered to be a comparison criterion. However, for the overall and complete controller comparison, the percent overshoot is taken into account.

Considering the *no load* test results, all three controllers had zero steady state error and no overshoot. The Incremental Controller tuning process was the easiest (one tuning parameter), but showed instability to large disturbances. The NIC and the LADRC proved to have better disturbance rejection properties as, on average, a faster recovery time to external disturbances was observed. Notice however, that the control signal for the NIC did not exhibit large voltage spikes in response to large disturbances. This agrees with the controller design criteria detailed in Section 4.4, where one of the design goals was to prevent large errors altogether. In addition, the control signal's noise frequency for the NIC was less when compared with the other two controllers.

The tabulated results for the *under load* tests are displayed in Table II. Analyzing these results, one can conclude that the Incremental Controller, in its original form, did not perform as well as the other two controllers due to the tendency of the integrator to wind-up. The NIC showed improved disturbance rejection properties, when compared

with the Incremental Controller, but the system response had overshoot and some steady state error. The advantage, however, consists of its ability to deal with stiction while providing good disturbance rejection. It also allowed the use of a 1st order pre-filter motion profile instead of a second order, which improved the control rate at low speeds (less computation time). In addition, its control signal does not have large voltage spikes in response to external disturbances, which prevents hardware (actuator) wear. One other benefit is its ability to prevent the development of large errors; thus improving the overall stability of the system.

However, as an overall comparison, the LADRC outperformed the other two controllers. Although the beginning part of the transient response had a relatively small oscillation, due to stiction and the delay in the actuator response time, the LADRC had zero overshoot and no steady state error. In terms of disturbance rejection, its capabilities are comparable with the NIC. One other important property to note was its control signal for the *under load* tests, as results showed that it was less noisy in both amplitude and frequency. The controller computation time was higher for this controller, which meant that its control update rate of 1.2 milliseconds was lower than the control rate for the other two controllers. Therefore, even with a low control update rate of ≈ 833 Hertz, the LADRC proved to be a robust controller as it was less sensitive to the system's dynamic variations, i.e. friction.

TABLE II: CONTROLLER COMPARISON (UNDER LOAD)

Performance	Controller		
	Incremental Controller	LADRC	Nonlinear Incremental Controller
Steady State Error (RPM)	0	0	10 – 30
Disturbance Rejection: Recovery Time (seconds)	$\approx 100\text{sec}$	$\approx 40\text{sec}$	$\approx 80\text{sec}$
% Overshoot (RPM)	0	0	0.067
Computation Time (microseconds)	$98\mu s$	$450\mu s$	$110\mu s$
Tuning Parameters	$k_i = 40$	$k_p = 50$ $\omega_o = 100$ $b_0 = 2$	$u_{\max} = 12000$ $\alpha = 0.1$ $\beta = 0.1$

CHAPTER VI

CONCLUDING REMARKS

Based on the experimental results presented in Section 5.4, it was verified that by treating friction as a disturbance to the desired system response, within the framework of ADRC, the nonlinear behavior of friction was successfully estimated and compensated in real time without the need of a friction model. Therefore using the form of the problem, this thesis provides a new solution to the vast realm of friction compensation where the performance of many friction compensation techniques is limited and dependent on the accuracy of the friction model.

In addition, due to the manufacturing problem, the control task became much more difficult due to the electrical and mechanical constraints mentioned in Section 1.2. These constraints promoted the development of a new, nonlinear-based, Incremental Controller (the NIC) which does not take much computation time, is intuitive to tune (can be parameterized to one tuning parameter), and is ideal for applications that have a high static friction coefficient, limited control effort or applications with flexible coupling. At the moment this controller is at its infancy, and additional research is needed to better understand and improve its capability.

Although the overall performance of the three controllers is comparable, they have shown good disturbance rejection properties. Interestingly enough, the integral control structure is common to all three controllers (refer to Chapter 4). Thus, one can also conclude that the integral control structure in itself provides good disturbance rejection capabilities. The trade off, however, consists of the additional pole and phase lag in the control loop, which in turn affects a system's stability margins.

The work of this thesis, in addition to the new contribution to the field of friction compensation, also includes the successful electrical engineering design of the Attitude Determination and Control Subsystem for the first lab based satellite at CSU (VikSat1). The design process included a variety of electrical and mechanical constraints, and different solutions to these limitations were found and implemented.

6.1 Future Research

The nonlinearity nature of friction affects the performance of many mechanical systems and thus influences a system's safety, performance, quality, and cost. Although the type of friction addressed in this thesis is that of dry rolling friction, it is the stepping stone towards the development of a generalized friction compensation controller that is not dependent on the friction model. Therefore as future research, this controller can be generalized, and then tested or simulated using the system addressed in this thesis or other systems along with different friction models such as the ones described in Section 2.2.

In addition, the ADRC framework can be further utilized by extending the ESO in order to obtain, otherwise unknown, information about the nonlinear frictional behavior. Depending on the specific application, this information can be used to improve the controller's performance and thus better compensate the undesired friction.

REFERENCES

1. E. Rabinowicz, *Friction and Wear of Materials*, 2nd ed., John Wiley & Sons Inc., 1995.
2. B. N. J. Persson, *Sliding Friction: Physical Principles and Applications*, Berlin; New York: Springer, 1998.
3. H. Olson, K. J. Astrom, C. Canudas de Wit, M. Gafvert, and P. Lischinsky, "Friction models and friction compensation," *Eur. J. Control*, Vol. 4, No. 3, pp. 176-195, 1998.
4. M. Janeic, "Friction compensation by the use of friction observer," Master Thesis, Lund Institute of Technology, 2004.
5. B. Armstrong-Helouvry, P. Dupont, and C. Canduas de Wit, "A survey of models, analysis tools and compensation methods for the control of machines with friction," *Automatica*, Vol. 30, No. 7, pp. 1083-1138, 1994.
6. Merriam-Webster Inc., *The Merriam-Webster Dictionary*, Springfield, Mass, 2004.
7. J. Larsen-Base, "Basic theory of solid friction," *ASM Handbook*, Volume 18: Friction, Lubrication and Wear Technology, ASM International, October, 1992.
8. A. Gemant, *Frictional Phenomena*, Chemical Pub. Co., Brooklyn, NY, 1950.
9. F. P. Bowden, and D Tabor, *Friction and Lubrication*, London, Methuen & Co. Ltd, 1956.
10. V. Hayward, and B. Armstrong, "A new computational method of friction applied to haptic rendering," *Experimental Robotics VI*, Springer: New-York, LNCS 250, pp. 404-412, 2000.
11. B. Armstrong-Helouvry, "Stick slip and control in low-speed motion," *IEEE Transactions on Automatic Control*, Vol. 38, No. 10, October, 1993.
12. B. J. Hamrock, and D. Dowson, *Ball Bearing Lubrication: The Elastohydrodynamics of Elliptical Contacts*, John Wiley & Sons Inc., 1981.
13. W. A. Glaeser, *Characterization of Tribological Materials*, Boston: Butterworth-Heinemann; Greenwich: Manning, 1993.
14. T. A. Harris, "Friction and wear of rolling-element bearings," *ASM Handbook*, Volume 18: Friction, Lubrication and Wear Technology, ASM International, October, 1992.

15. D. Dowson, *History of Tribology*, London; New York: Longman, 1979.
16. G. A. Tomlinson, "A molecular theory of friction," *Phil. Mag. Series 7*, pp. 905-939, 1929.
17. D. Tabor, "The mechanism of rolling friction. II. The elastic range," *Proc. R. Soc. London. Series A, Mathematical and Physical Sciences*, Vol. 229, No. 1177, pp. 198-220, 1955.
18. B. J. Hamrock, S. R. Schmid, and Bo O. Jacobson, *Fundamentals of Fluid Film Lubrication*, 2nd ed., New York, NY, Marcel Dekker Inc., 2004.
19. B. Armstrong, and C. Canudas de Wit, "Friction modeling and compensation," *The Control Handbook*, W. S. Levine, CRC Press 1996.
20. R. H. A. Hensen, "Controlled mechanical systems with friction," PhD Dissertation, Technische Universiteit Eindhoven, The Netherlands, 2002.
21. P. A. Bliman, "Mathematical study of the Dahl's friction model," *European Journal of Mechanic, A/Solids*, Vol. 11, No. 6, pp. 835-848, 1992.
22. C. Canudas de Wit, H. Olsson, K. J. Åström, and P. Lischinsky, "A new model for control of systems with friction," *IEEE Transactions on Automatic Control*, Vol. 40, No. 3, March, 1995.
23. C. Canudas de Wit, and P. Lischinsky. "Adaptive friction compensation with partially known dynamic friction model," *Int. Journal of Adaptive Control and Signal Processing* (Special Issue on Adaptive Systems with Nonsmooth Nonlinearities), Vol. 11, Issue 1, pp. 65-80, 1997.
24. P. R. Dahl, "Solid friction damping of mechanical vibrations," *AIAA Journal*, Vol. 14, No. 12, pp. 1675-82, 1976.
25. J. Han, "Nonlinear design methods for control systems," *The Proc. Of The 14th IFAC World Congress*, Beijing, 1999.
26. Z. Gao, "Active disturbance rejection control: A paradigm shift in feedback control system design," *Proceedings of the 2006 American Control Conference*, Minneapolis, Minnesota, June 14-16, 2006.
27. K. J. Astrom, and T. Hagglund, "PID Control," *The Control Handbook*, W. S. Levine, Ed. P. 198, CRC Press 1996.
28. J. R. Wertz, *Spacecraft Attitude Determination and Control*, Dordrecht, Holland, D. Reidel Publishing Company, 1978.
29. J. R. Wertz, and W. J. Larson, *Space Mission Analysis and Design*, Torrance, Calif: Microcosm; Dordrecht; Boston: Kluwer, 1999.

30. V. L. Piscane, *Fundamentals of Space Systems*, New York, New York, Oxford University Press. Inc., 2005.
31. M. D. Griffin, and J. R. French, *Space Vehicle Design*, Washington, DC: American Institute of Aeronautics and Astronautics, 1991.
32. P. Yedamale, "Brushless DC (BLDC) motor fundamentals," Microchip Application Note: AN885, [Online]. Available: www.microchip.com.
33. NASA (Unknown), Practice No. PD-ED-1229, [Online]. Available: http://klabs.org/DEI/References/design_guidelines/design_series/1229msfc.pdf#search='selection%20of%20electric%20motors%20for%20aerospace%20applications'.
34. R. Condit, and D. W. Jones, "Stepping motors fundamentals," Microchip Application Note: AN907, [Online]. Available: www.microchip.com.
35. J. Sprunck, "Spacecraft motor selection," Article can no longer be found.
36. BEI – Kimco Magnetic Division, "Brushless DC motors: An application guide," [Online]. Available: <http://www.beikimco.com/appguides/motors.php>.
37. Maxon Precision Motors, Inc., [Online]. Available: www.maxonmotorusa.com.
38. Danaher Motion Product Line, "Overview of feedback device," [Online]. Available: <http://www.pacsci.com/support/documents/mtd/d-21.pdf#search='overview%20of%20feedback%20devices'>.
39. T. E. Kissell, *Industrial Electronics: applications for programmable controllers, instrumentation and process control, and electrical machines and motor control*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 2000.
40. FAQ Guide, [Online]. Available: www.usdigital.com.
41. Danaher Motion, "Servo handbook: velocity resolution issues," [Online]. Available: http://www.danahermotion.com/education/learn_about_mc/servohandbook/feedback/systemresolution.php.
42. K. Gauen, and J. Alberkrack, "Three piece solution for brushless motor controller design," ON Semiconductor Application Note: AN1046, [Online]. Available: <http://onsemi.com>.
43. A. Bloom, "Making sense of sun sensors," [Online]. Available: www.amsat.org.
44. J. R. Zeller, "Technology issues related to modern motion control," Presentation Handout, Cleveland State University.
45. B. C. Kuo, and F. G. Golnaraghi, *Automatic Control System*, John Wiley & Sons, Inc., 2003.

46. J. Han, "Auto-disturbance rejection control and its applications", *Control and Decision*, Vol. 13, No. 1, pp.19-23, 1998. (In Chinese)
47. R. Miklosovic, "Development of a robust framework for controlling high performance turbofan engines," Doctoral Dissertation, Cleveland State University, May 2006.
48. Z. Gao, Y. Huang, and J. Han, "An alternative paradigm for control system design," *Proc. Of the 2001 IEEE Conference on Decision and Control*, Dec. 2001.
49. Z. Gao, "Scaling and bandwidth-parameterization based controller tuning," *Proc. of the American Control Conference*, pp. 4989-4996, 2003.
50. J. Han, "Control theory: Is it a theory of model or control?," *Systems Science and Mathematical Sciences*, Vol. 9, No. 4, pp.328-335, 1989. (In Chinese)
51. Z. Gao, "From linear to nonlinear control means: A practical progression," *ISA Transactions*, 2001.
52. R. Miklosovic, and Z. Gao, "A dynamic decoupling method for controlling high performance turbofan engines," *Proc. of the 16th IFAC World Congress*, Prague, Czech Republic, July 2005.
53. Z. Gao, S. Hu, and F. Jiang, "A novel motion control design approach based on active disturbance rejection", *IEEE Conference on Decision and Control*, Vol. 5, pp. 4578-4585, 2001.
54. B. Fast, "Form based control and rate approximation of a MEMS gyroscope", Doctoral Dissertation, Cleveland State University, December 2006.
55. A. Radke, and Z. Gao, "A survey of state and disturbance observers for practitioners", *American Control Conference*, June 14-16, 2006.
56. R. Miklosovic, A. Radke, and Z. Gao, "Discrete implementation and generalization of the extended state observer", *American Control Conference*, June 14-16, 2006.
57. N. Tollis, B. Fast, R. Miklosovic, and P. Chu "A high performance controls processing architecture," To be published.
58. G. F. Franklin, J. D. Powell, and M. Worksman, *Digital Control of Dynamic System*, 3rd ed., Menlo Park, CA: Addison Wesley Longman, Inc., 1998.
59. K. Astrom, and T. Hagglund, "PID Controllers: Theory, design and tuning," *North Carolina: Instrument Society of America*, 1995.
60. Z. Gao, and S. Hu, "On proprieties and applications of a new form of discrete time optimal control law," *Industry Applications Conference*, Vol. 3, pp. 1511-158, October 3-7, 2004.

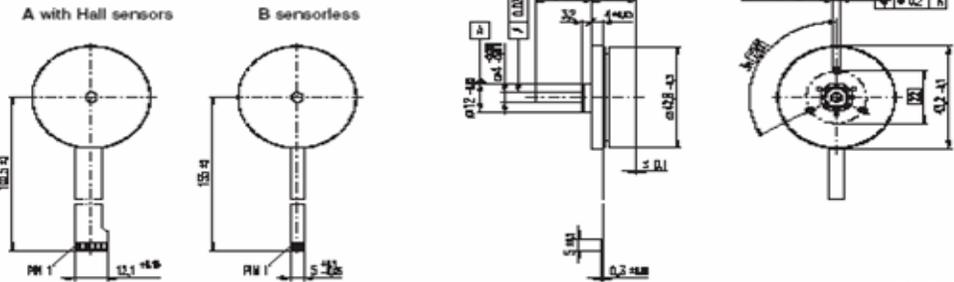
61. Z. Gao, "On discrete time optimal control: A closed-form solution," *Proceedings of the American Control Conference*, Boston, MA, June 30 - July 2, 2004.
62. R. Miklosovic, "A generalization of the PID design framework", To Be Published.
63. A. Radke, "A java based interactive control design and tuning platform", Master Thesis, Cleveland State University, May 2002.
64. Q. Zheng, and Z. Gao, "Motion control design optimization: Problem and solutions", *International Journal of Intelligent Control and Systems*, Vol. 10, No. 4, December 2005, 269-276.

APPENDICES

A. Momentum Wheel Motor Datasheet

maxon flat motor

EC 45 Flat motor Ø45 mm, brushless, 30 Watt



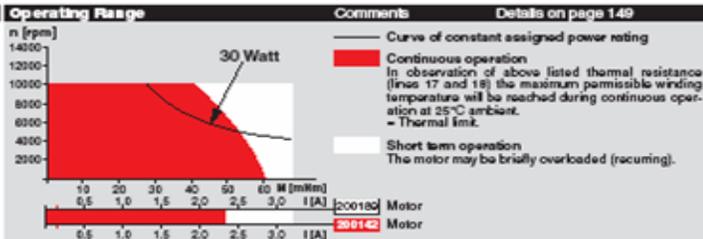
- Stock program
- Standard program
- Special program (on request!)

Order Number

Motor Data	with Hall sensors		sensorless	
	200142	200180	200180	200142
1 Assigned power rating	W	30	30	
2 Nominal voltage	Volt	12.0	12.0	
3 No load speed	rpm	4400	4400	
4 Stall torque	mNm	260	220	
5 Speed / torque gradient	rpm / mNm	17.5	20.5	
6 No load current	mA	150	150	
7 Terminal resistance phase to phase	Ohm	1.20	1.40	
8 Max. permissible speed	rpm	10000	10000	
9 Max. continuous current at 5000 rpm	A	2.30	2.30	
10 Max. continuous torque at 5000 rpm	mNm	53.8	53.8	
11 Max. efficiency	%	77.6	78.0	
12 Torque constant	mNm / A	25.5	25.5	
13 Speed constant	rpm / V	374	374	
14 Mechanical time constant	ms	17.0	20.0	
15 Rotor inertia	gm ²	92.5	92.5	
16 Terminal inductance phase to phase	mH	0.410	0.410	
17 Thermal resistance housing-ambient	K / W	4.0	4.0	
18 Thermal resistance winding-housing	K / W	6.0	6.0	
19 Thermal time constant windings	s	11.6	11.6	
20 Thermal time constant stator	s	42.2	42.2	

- Specifications**
- Axial preload > 5 N
 - Max. **ball bearing** loads
 - axial (dynamic) 2.8 N
 - radial (7.5 mm from flange) 5.5 N
 - Force for press fits (static) 50 N (static, shaft supported)
 - Ambient temperature range -40 ... +100°C
 - Max. permissible winding temperature +125°C
 - Weight of motor 88 g
 - Version with and without Hall sensors
 - 16 pole permanent magnet
 - 3 phased coil stator with 4 pole shoes each
 - Values listed in the table are nominal.
 - Connections with Hall sensors sensorless

Pin 1	4.5 ... 18 VDC	Motor winding 1
Pin 2	Hall sensor 3*	Motor winding 2
Pin 3	Hall sensor 1*	Motor winding 3
Pin 4	Hall sensor 2*	neutral point
Pin 5	GND	
Pin 6	Motor winding 3	
Pin 7	Motor winding 2	
Pin 8	Motor winding 1	
 - Adapter Order Number Order Number
see p. 266 220300 220310
 - Connector Article number Article number
AMP 1-487051-1 487051-4
MOLEX 52207-1190 52207-0490
MOLEX 52080-1110 52080-0410
 - Pin for design with Hall sensors:
FPC, 11 pins, pitch 1.0 mm, top contact style
 - Internal pull-up (7 ... 18 kΩ) on pin 1
 - For wiring diagram for Hall sensors, see p. 26



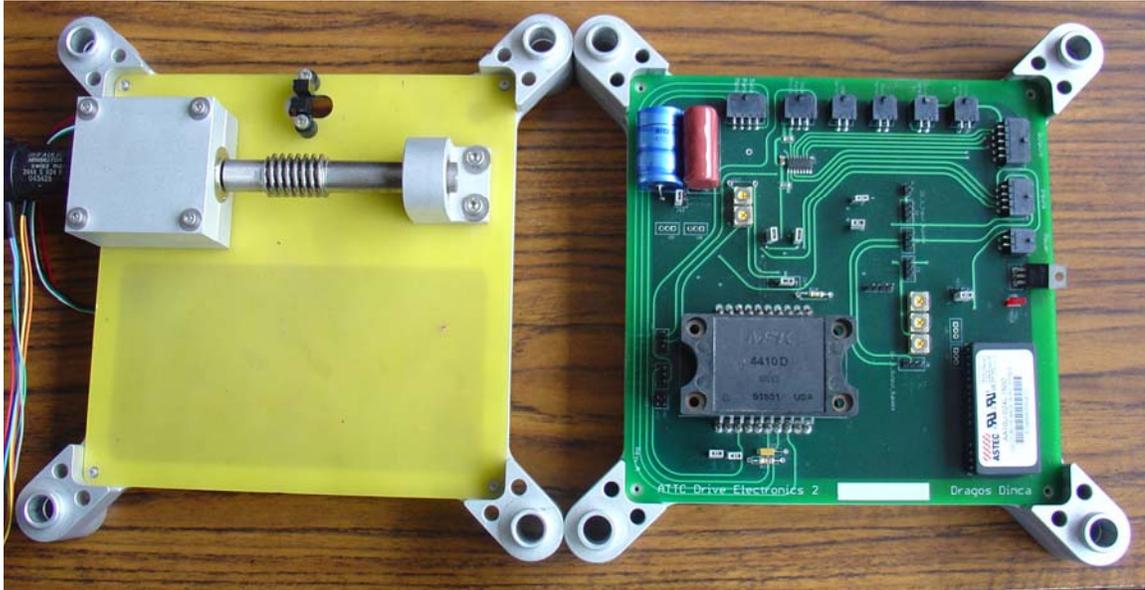
maxon Modular System Overview on page 17 - 21

Spur Gearhead
Ø45 mm
0.5 - 2.0 Nm
Details page 220

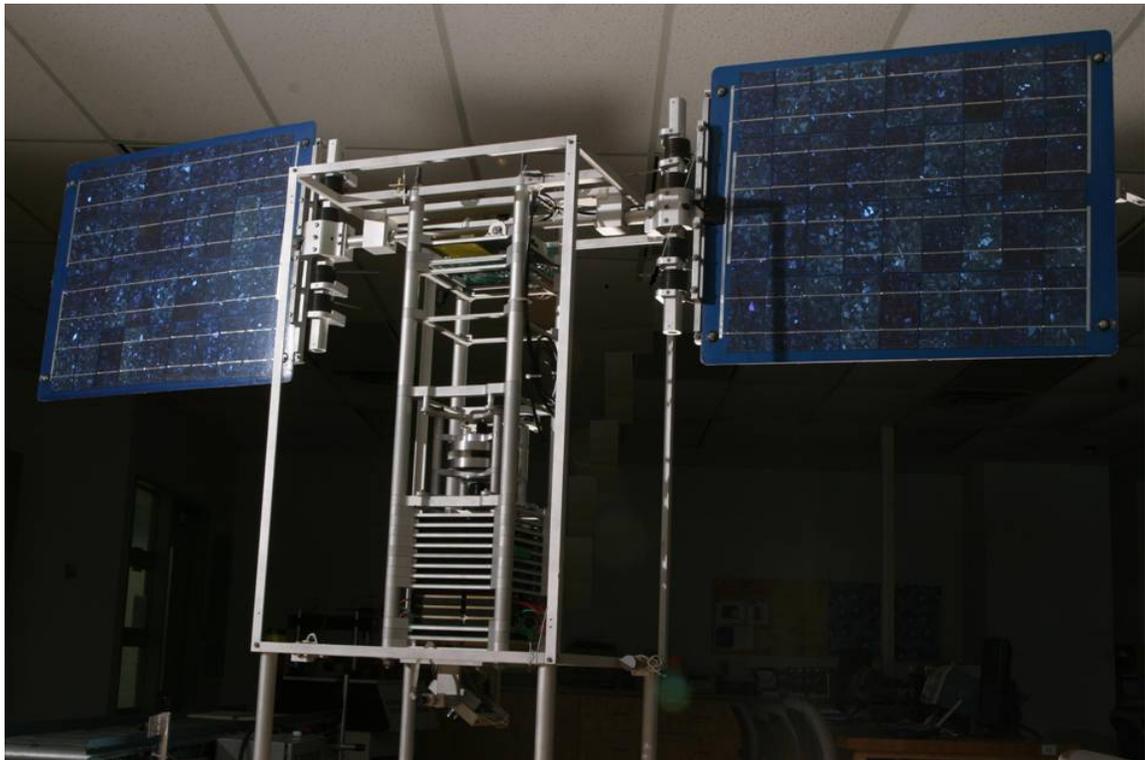
Recommended Electronics:
AEC5 36/3 page 256
DEC 50/5 25.8
Notes 17

July 2004 edition / subject to change

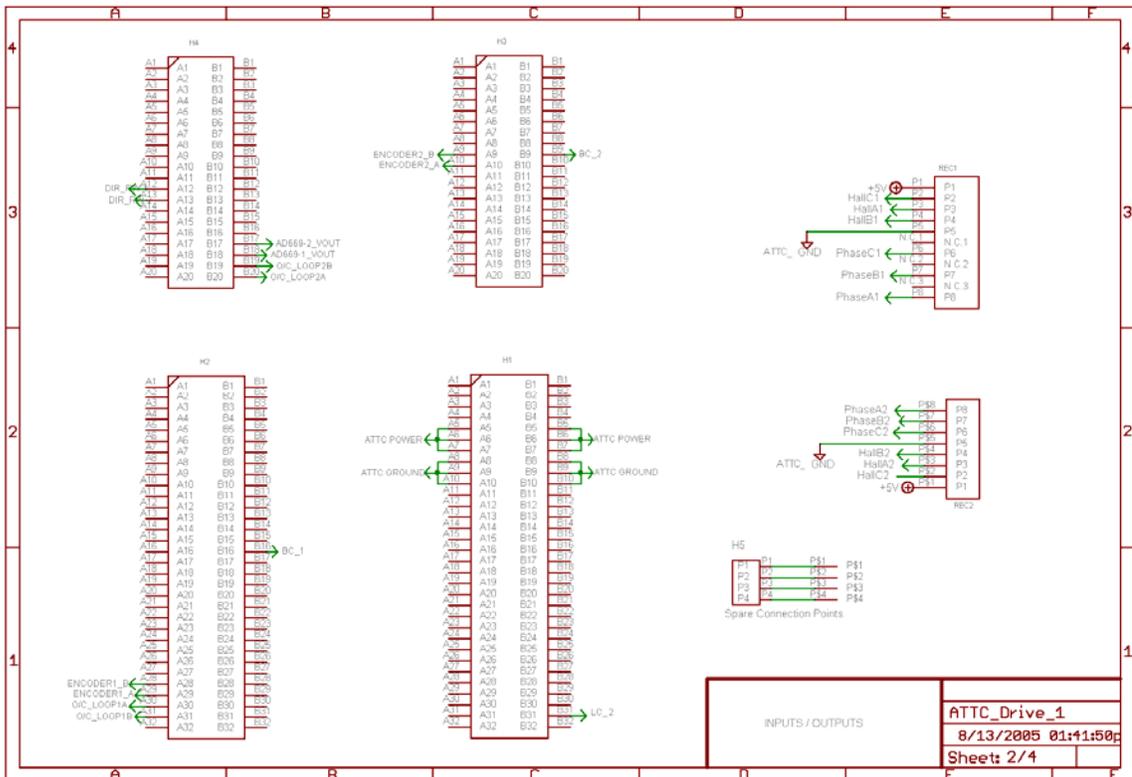
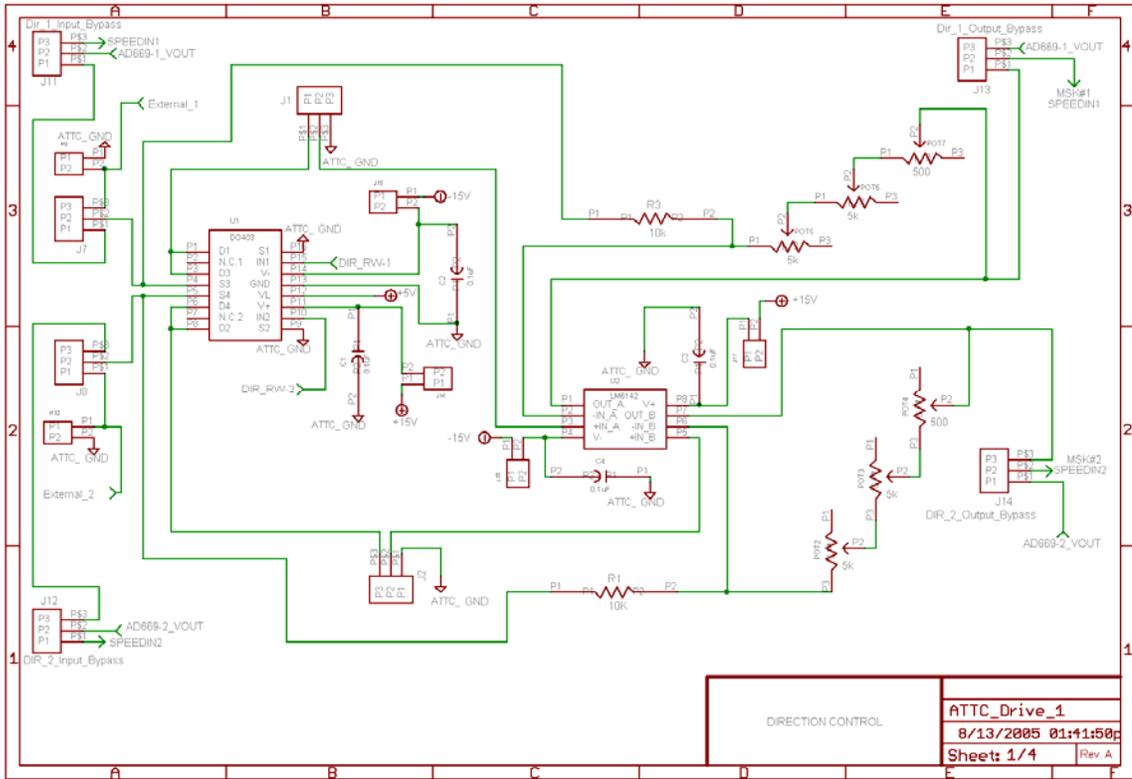
B. Solar Panels Control PCB

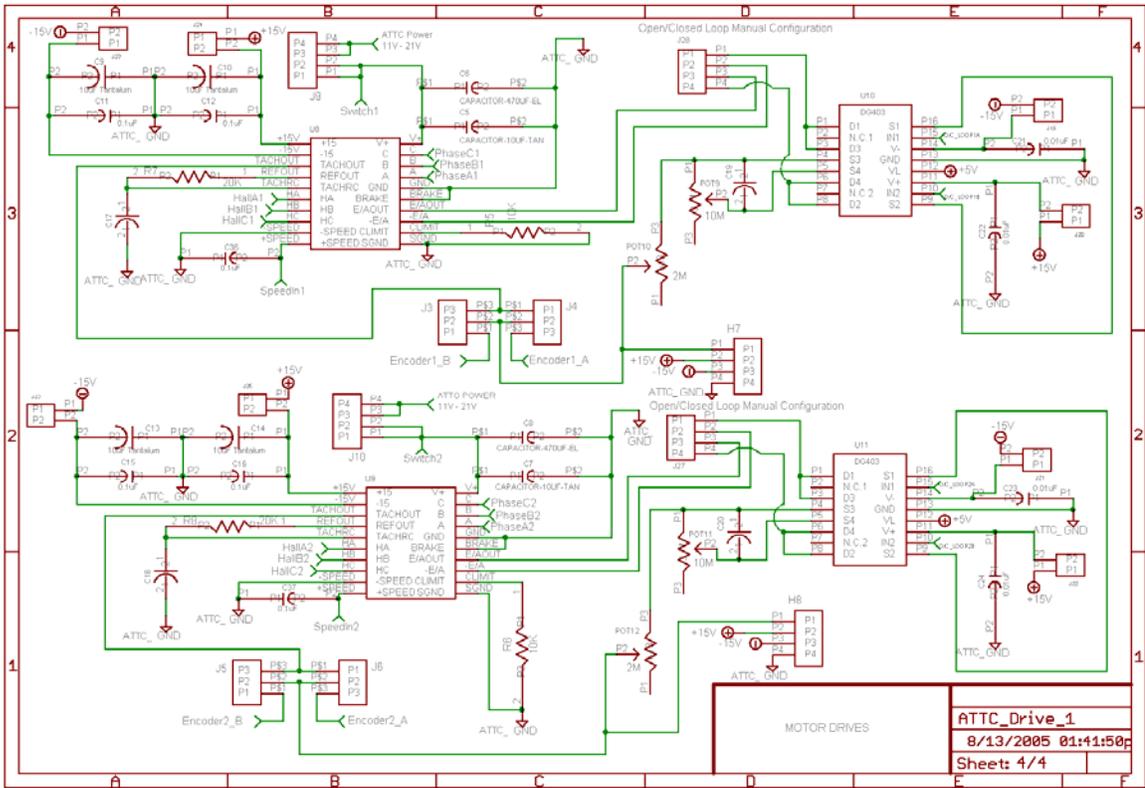
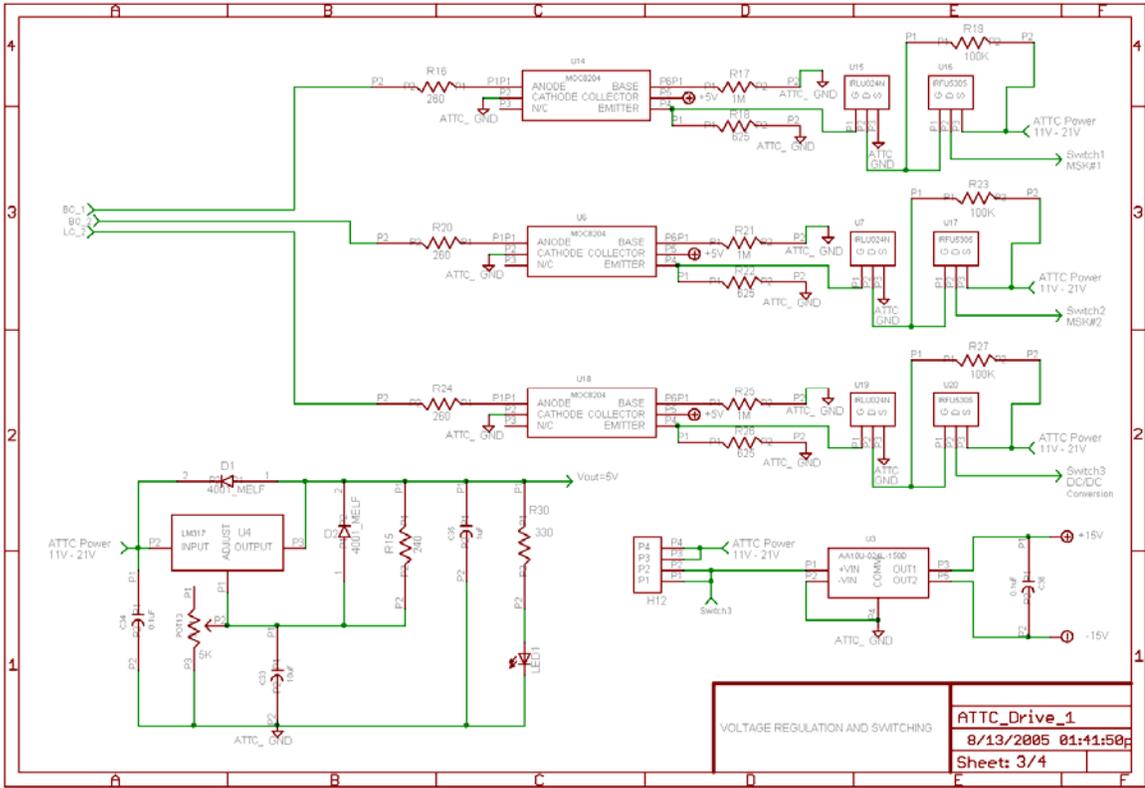


C. VikSat1



D. Drive Electronics PCB – Design Schematics





E. ADCS Microcontroller PCB – Design Schematics

