Adaptive Control Applied to the Cal Poly Spacecraft Attitude Dynamics Simulator

A Thesis
Presented to the Faculty of
California Polytechnic State University, San Luis Obispo

In Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Aerospace Engineering

By
Matthew C Downs

October, 2009

© 2009
Matthew C. Downs
ALL RIGHTS RESERVED

Committee Membership

Title: *Adaptive Control Applied to the Cal Poly Spacecraft Attitude Dynamics Simulator*

Author: Matthew C Downs

Date Submitted: October, 2009

Committee Chair: Dr. Eric Mehiel
Assistant Professor/ Department Chair
Cal Poly Aerospace Engineering

Committee Member: Dr. Jordi Puig-Suari
Professor, Cal Poly Aerospace Engineering

Committee Member: Dave Esposto
Lecturer, Cal Poly Aerospace Engineering

Committee Member: Dr. Charles Birdsong
Assistant Professor, Cal Poly Mechanical Engineering

Abstract

Adaptive Control Applied to the Cal Poly Spacecraft Attitude Dynamics Simulator

Matthew C. Downs

The goal of this thesis is to use the Cal Poly Spacecraft Attitude Dynamics Simulator to provide proof of concept of two adaptive control theories developed by former Cal Poly students: Nonlinear Direct Model Reference Adaptive Control and Adaptive Output Feedback Control. The Spacecraft Attitude Dynamics Simulator is a student-built air bearing spacecraft simulator controlled by four reaction wheels in a pyramidal arrangement. Tests were performed to determine the effectiveness of the two adaptive control theories under nominal operating conditions, a “plug-and-play” spacecraft scenario, and under simulated actuator damage. Proof of concept of the adaptive control theories applied to attitude control of a spacecraft is provided. The adaptive control theories are shown to attain similar or improved performance over a Full State Feedback controller. However, the measurement capabilities of the simulator need to be improved before strong comparisons between the adaptive controllers and Full State Feedback can be achieved.

Acknowledgments

I Would Like to Thank:

My Advisor
Dr. Eric Mehiel
for
His Support During This Project

My Parents
Mike and Resa Downs
for
Funding My Education

And

My Fiancé
Jen Ross
for
Her Patience and Support

Table of Contents

List of Figures.....	viii
Chapter I: Introduction.....	1
1.1 The Cal Poly Spacecraft Attitude Dynamics Simulator.....	1
1.2 Adaptive Control.....	3
1.3 Thesis Objectives and Overview.....	5
Chapter II: Spacecraft Attitude Dynamics Simulator Overview.....	7
2.1 Pyramidal Reaction Wheel Platform.....	7
2.2 Subsystem Hardware.....	7
2.2.1 Measurement.....	8
2.2.2 Reaction Wheels.....	9
2.2.3 Power, Air Bearing, and Mass Balancing System.....	9
2.2.4 Data Acquisition and Control Command.....	10
2.2.5 Data Acquisition and Control Command Interface Electronics.....	11
2.3 Spacecraft Attitude Dynamics Simulator Software.....	14
Chapter III: SADS Dynamics and Torque Control.....	16
3.1 PRWP Geometry.....	16
3.2 Torque Generation.....	18
3.3 Quaternion Kinematics.....	21
3.4 Reaction Wheels.....	22
Chapter IV: PRWP Attitude Control.....	27
4.1 Full State Feedback Control Overview.....	27
4.2 NDMRAC and AOF Overview.....	28

4.3 Test Cases.....	30
4.4 Implementation of FSFB.....	33
4.5 Implementation of NDMRAC and AOF.....	34
4.6 Computer Model of Simulator.....	35
Chapter V: Simulation Results.....	37
5.1 Desired Quaternion.....	37
5.2 Computer Simulation Results.....	37
5.3 SADS Simulation Results.....	42
Chapter VI: Conclusion.....	51
6.1 Computer and SADS Simulation Conclusions.....	51
6.2 Future Recommendations.....	53
List of References.....	55
Appendix A: Matlab Code.....	57
Appendix B: Simulink Block Diagrams.....	66
Appendix C: DACC Interface Electronics Schematics.....	90

List of Figures

Figure 1.1 PRWP in 2008.....	2
Figure 1.2 Current PRWP.....	3
Figure 2.1 SADS Subsystems.....	8
Figure 2.2 DACC Interface Electronics.....	12
Figure 3.1 PRWP Reference Frames.....	16
Figure 3.2 Reaction Wheel Geometry.....	17
Figure 3.3 Wheel Speed Measurement Noise.....	23
Figure 3.4 Wheel Speed Kalman Filter.....	24
Figure 4.1 Generic Direct Adaptive Model Reference Controller.....	28
Figure 4.2 Solid Model of the PRWP.....	31
Figure 4.3 Nominal and Degraded Reaction Wheel Speeds.....	32
Figure 4.4 Reference Model.....	34
Figure 4.5 Computer Model of Simulator.....	35
Figure 5.1 Computer Simulation of First Test Case.....	38
Figure 5.2 Computer Simulation of Second Test Case.....	39
Figure 5.3 Computer Simulation of Third Test Case.....	40
Figure 5.4 Computer Simulated Body Torques for Third Test Case.....	41
Figure 5.5 SADS Simulation of First Test Case.....	42
Figure 5.6 Zoom View of SADS First Test Case.....	43
Figure 5.7 SADS Simulation of Second Test Case.....	45
Figure 5.8 SADS Simulation of Third Test Case.....	46
Figure 5.9 NDMRAC and AOF Third Test Body Torques.....	47

Figure 5.10 NDMRAC and AOF with Decreased Adaptive Parameters.....	49
--	----

Chapter I: Introduction

Air bearing dynamics simulators provide a low cost method of simulating dynamics in a torque-free environment. They find wide usage as spacecraft dynamics simulators because spacecraft generally operate under low external torques. For the past few years, Cal Poly has been developing the Cal Poly Spacecraft Attitude Dynamics Simulator (SADS), a reaction wheel controlled air bearing simulator. One of the design goals of the SADS project is to be able to use it to easily test and verify spacecraft attitude control laws. This thesis will use the SADS to provide proof of concept of two adaptive control laws applied to spacecraft attitude dynamics control.

1.1 The Cal Poly Spacecraft Attitude Dynamics Simulator

The Cal Poly SADS project began in 2002 with the creation of a simple air bearing platform¹. In 2006, Cal Poly Mechanical Engineering students rebuilt the air bearing platform as the Pyramidal Reaction Wheel Platform (PRWP) depicted in Figure 1.1². Mittelsteadt developed the original sensor and actuator design of the PRWP and Healy developed a system identification algorithm^{3,4}. In 2007, Saile developed the fine balance system and Liu et al. developed wireless communication between the PRWP and a “ground” computer^{5,6}. In 2008, Logan improved upon the control and sensing of the PRWP and Silva demonstrated proof of concept of Healy’s system identification algorithm^{7,8}. Figure 1.1 shows the PRWP as it was in 2008.

Figure 1.1 PRWP in 2008⁸

Wireless command and control of the PRWP was accomplished using a Gumstix/Robostix system. Gumstix and Robostix are small single board computers. The Gumstix computer's main purpose was to communicate wirelessly with a "ground" computer via a Bluetooth connection. The Robostix computer sent and received signals to and from the actuators and sensors. However, due to limitations of the Gumstix/Robostix system, system control could only be performed at a rate of 1 Hz, resulting in poor attitude control performance⁷. For the SADS to function as a test bed for spacecraft attitude control laws, the control rate needed to be increased significantly. In order to remedy this, Kinnett replaced the Gumstix/Robostix system with a PC/104 based on-board computer and data acquisition device and Downs and Kinnett created circuits to interface between the sensors and actuators and the data acquisition device. With the PC/104 on-board computer, the SADS is capable of control at a rate of 50 Hz. Figure 1.2 shows the PRWP in its current state.

Figure 1.2 Current PRWP

The PRWP is discussed in more detail in Chapter II.

1.2 Adaptive Control

Adaptive control is receiving increased attention in aerospace engineering because of adaptive controllers' ability to cope with complex plant dynamics that are changing or uncertain. Adaptive controllers change the control law in order to account for these changes and uncertainties. A common example of plant dynamics that change in a complex way is an aircraft that has suffered significant structural damage to an aerodynamic surface such as losing a tail. An example of uncertain plant dynamics is an aircraft flying at a high Mach number or a high angle of attack where aerodynamic effects are uncertain. There is currently great interest in adaptive control applied to damaged aircraft and aircraft with complex flight dynamics. Nguyen et al. investigated the use of adaptive control in order to regain control of a damaged generic transport vehicle and asymmetric aircraft and Liu et al. investigated the use of adaptive control in the stability recovery of

aircraft with control surface failures^{9,10}. These are examples of adaptive control applied to changing plant dynamics. Shin et al. investigated adaptive control applied to fighter aircraft flying in highly non-linear flight regimes and Lee et al. investigated adaptive control applied to aircraft with unknown flight parameters^{11,12}. These are examples of adaptive control applied to uncertain plant dynamics.

Spacecraft are also subject to changing and uncertain plant dynamics. The mass properties of a spacecraft change when fuel is burned, mass is jettisoned, or rendezvous occurs. Fuel slosh and flexible structures introduce very complex dynamics into the spacecraft plant. Shageer et al. investigated adaptive control applied to fuel sloshing during a spacecraft maneuver and Kharisov et al. investigated the use of adaptive control to improve the safety of flexible launch vehicles^{13,14}. Other possible applications of adaptive control to spacecraft are in modular or “Plug-and-Play” spacecraft and spacecraft with actuator failures. The concept of a modular or “Plug-and-Play” spacecraft is that spacecraft components and subsystems are preconfigured to work with each other. A major advantage of this concept is that spacecraft could be designed, built, and launched very quickly. It would be advantageous to not waste time on the ground or on orbit performing mass property identification. In this case, the spacecraft’s dynamics would be unknown. An actuator failure could also be the cause of changing plant dynamics. For example, an actuator failure for a spacecraft may involve the loss or degraded use of one or more reaction wheels. Both Scarritt and Harvey developed direct adaptive control laws and investigated

the use of these control laws for attitude control of “Plug-and-Play” spacecraft^{15,16}.

Two former graduate students at Cal Poly developed adaptive control laws and demonstrated the application of those control laws to spacecraft attitude control via computer simulations. Torres developed Nonlinear Direct Model Reference Adaptive Control (NDMRAC) and Patel developed Adaptive Output Feedback (AOF)^{17,18}. Both NDMRAC and AOF were applied to the rigid body equations of motion. In this thesis, proof of concept of NDMRAC and AOF applied to the attitude control of a spacecraft will be shown through SADS simulations.

1.3 Thesis Objectives and Overview

The ultimate goal of this thesis is to use the SADS to establish proof of concept of NDMRAC and AOF for the attitude control of spacecraft with unknown mass properties or damaged actuators. The performance of NDMRAC and AOF in these areas will be compared to Full State Feedback.

Chapter II provides an overview of the SADS. The changes that were made to the SADS in order to improve controllability will be discussed.

Chapter III discusses the SADS dynamics, how torque is generated by the reaction wheels, and quaternion kinematics.

Chapter IV presents Full State Feedback, NDMRAC, and AOF controllers and how they are implemented.

Chapter V presents and discusses the results of the computer and SADS simulations. The NDMRAC and AOF controllers are shown to be effective when applied to the attitude control of a rigid body.

Chapter VI provides conclusions and future recommendations.

Chapter II: Spacecraft Attitude Dynamics Simulator Overview

2.1 Pyramidal Reaction Wheel Platform

The Pyramidal Reaction Wheel Platform, PRWP, and the hardware and software installed on it compose Cal Poly's Spacecraft Attitude Dynamics Simulator (SADS). The PRWP consists of an aluminum chassis supported by a hemispherical air bearing. The chassis has an upper and lower deck which are used to mount subsystem components. The chassis also provides attachment points for the four reaction wheels. Six coarse mass balances are located in channels and can be used to balance the platform by hand. There is also a fine mass balancing system that can be used to balance the platform very accurately. For more information on the design and configuration of the PRWP, the reader is referred to Mittelsteadt, Logan, Silva, and Kinnett^{3,7,8,19}. The next section will discuss the subsystem hardware.

2.2 Subsystem Hardware

The Spacecraft Attitude Dynamics Simulator has seven main subsystems. These include the air bearing, chassis, power, measurement sensors, data acquisition and control command, balancing, and reaction wheels. A schematic of the how these subsystems interact is shown in Figure 2.1.

Figure 2.1 SADS Subsystems

2.2.1 Measurement

There are a total of seven sensors mounted on the PRWP, including three microelectromechanical systems (MEMS) gyroscopes, and four motor encoders. Eventually, an LN-200 high performance gyroscope will be installed as well. The MEMS gyroscopes are Silicon Sensing CRS03-01S single axis gyroscopes. The MEMS gyroscopes have a high drift rate of $\pm 0.55^\circ/\text{s}$ and are limited to angular rates less than $100^\circ/\text{s}$. The motor encoders are US Digital EM1 Transmissive Optical Encoder Modules with US Digital HUBDISK-1 codewheels with 32 counts per revolution. The codewheels are mounted to the motor shaft. The encoder modules read the codewheel and output a quadrature signal.

2.2.2 Reaction Wheels

Each reaction wheel consists of a wheel, DC motor, and motor driver. Each wheel is milled from aluminum and mounted to the motor's drive shaft. Mittelsteadt³ estimated that the wheels have a moment of inertia of 5.3552 kg-cm². The reaction wheel motors are Faulhaber 3863-024C DC-Micromotors. They are capable of speeds of up to 8,000 revolutions per minute and have a maximum torque of .11 Newton-meters. For more information on the wheels and motors, the reader is referred to Mittelsteadt, Logan, and Silva^{3,7,8}.

As Logan mentions in his thesis, previous versions of the motor drivers were susceptible to overheating at high current loads. As recommended, new Pololu MD05A MC33887 Motor Driver Carriers were installed. These motor driver carriers deliver a Pulse Width Modulated (PWM) voltage from the 24 Volt battery stack to the motors. They operate from 0 to 100% commanded PWM duty cycle and are capable of braking and providing reverse voltage so that the wheels can rotate clockwise or counterclockwise as well as accelerate or decelerate on command.

2.2.3 Power, Air Bearing, and Mass Balancing System

Power is supplied to the subsystem components via three separate battery stacks. Two rechargeable 12 Volt, 5 Amp-hour, lead-acid batteries in series provide 24 Volts to the reaction wheel and mass balancing subsystems. Two rechargeable 16.8 Volt, 4.2 Amp-hour Nickel-metal hydride batteries in series provide a nominal 36 Volts to the PC/104 computer and the Data Acquisition and Control Command (DACC) interface electronics. The PC/104's

DC/DC power supply regulates this supply down to 5 Volts. One of the batteries from the 36 Volt stack supplies 18 Volts to the DACC interface electronics where the voltage is regulated to 5 Volts, 2.5 Volts, and 12 Volts. These voltages power the MEMS gyroscopes, motor encoders, and other electronics.

The air bearing is hemispherical with the male side mounted on the bottom of the PRWP. An air compressor provides 60 psi air to the female side of the bearing which is mounted on a stand.

The mass balancing system consists of six coarse balance masses which are manually moved in tracks to balance the PRWP in three directions by hand. In addition, there are three fine mass balances that can precisely position the center of gravity of the PRWP by moving masses on lead screws with servomotors.

For more information about the power, air bearing, and fine mass balance system, the reader is referred to Mittelsteadt, Saile, Logan, and Silva^{3,5,7,8}.

2.2.4 Data Acquisition and Control Command

The Data Acquisition and Control Command subsystem is responsible for receiving measurement data, interpreting it, and sending commands to the actuators. This subsystem originally consisted of a Gumstix microcomputer with a Robostix robotics controller. This system sent measurement data to a “ground” computer which interpreted the information and sent commands back to the Gumstix/Robostix board via a Bluetooth wireless connection. However, this configuration was only capable of sending and receiving information at a rate of 1 Hz. The Gumstix/Robostix configuration was replaced with a PC/104 computer

and a series of circuit boards designed to be an interface between the sensors and actuators and the computer.

PC/104 is an embedded computer standard that allows computer components to be stacked into a compact and rugged form factor. The on-board PC/104 based computer stack consists of a motherboard, data acquisition board, serial communications board, DC/DC power supply, and a hard drive. The PC/104 motherboard is a Kontron MOPSPM104 with an Intel Pentium M processor and 512 MB RAM. Data acquisition is accomplished by an Eagle Technology PC104PLUS-30C data acquisition device with 16 analog inputs, 4 analog outputs, and 24 digital input/outputs. Currently, the digital input/output ports on the PC/104 data acquisition board are not functional. A National Instruments USB 6008 data acquisition board is used for digital outputs instead of the PC/104 data acquisition board. The serial communications board is an Xtreme/104-*Plus* with RS-485 communication capabilities. The computer stack is capable of wireless networking via a Linksys USB Wireless G adapter. A National Instruments USB 6008 data acquisition board is used for digital outputs instead of the PC/104 data acquisition board. For more information on the on-board computer stack, the reader is referred to Kinnett¹⁹.

2.2.5 Data Acquisition and Control Command Interface Electronics

The purpose of the DACC interface electronics is to provide power to some of the sensors, transform data from the sensors to a form that the data acquisition system can read, and to transform data from the control command system to a form that the actuators can use. As mentioned, the previous

configuration of the Data Acquisition and Control Command subsystem was only capable of proving control commands at 1 Hz, which is insufficient for everyday control demonstration and proof of concept of new control theories. This prompted the use of the PC/104 on board computer as an integrated data acquisition system, data interpretation system, and command generator. However, previous versions of the interface electronics that were designed to work with the Gumstix/Robostix system became obsolete. Therefore, new interface electronics became necessary. The new version of the interface electronics consists of a stack of circuit boards that are mounted onto the bottom deck of the PRWP. Figure 2.4 shows the interface electronics stack with each circuit board labeled.

Figure 2.2 DACC Interface Electronics

The DACC interface electronics stack consists of five circuit boards, each with a specific purpose.

The power board regulates the 18 Volt supply down to 5 Volts, 2.5 Volts, and 12 Volts. The 5 Volt supply provides power to the MEMS gyroscopes, motor drivers, motor encoders, wheel direction circuit, and PWM generation circuit. The 2.5 Volt and 12 Volt supplies power the frequency to voltage conversion circuit. Additionally, the power board provides reverse and overvoltage protection to the sensors and other electronics.

The wheel direction circuit is designed to route power to the motor encoders and determine the direction that the reaction wheels are spinning. Currently, the wheel direction circuit does not reliably determine the direction that the wheels are spinning. This does not pose a significant problem because the wheels are rotated in a known direction and are not permitted to rotate at angular velocities close to zero. In this way, the wheel direction is always known without the need for the wheel direction circuit.

The frequency to voltage conversion circuit acts as a tachometer for the wheels. It converts the frequency of one of the pulses of a motor encoder's quadrature output into a voltage proportional to that frequency. The design is a simplified and more compact version of the one presented by Logan. The frequency to voltage conversions for the four reaction wheels are performed by four LM2917 Frequency to Voltage Converter integrated circuits with a few additional components.

The PWM generation circuit converts four independent analog voltages from the analog outputs on the data acquisition board into four independent PWM signals with duty cycles determined by the values of the analog output voltages. Two op-amps generate a single triangle waveform, which is then compared to the analog output voltages by a quad op-comparator. The result is four PWM signals with commandable duty cycles. The PWM signals are sent to the motor drivers where the motor drivers deliver the same waveform, but from the 24 Volt source.

The signal routing circuit board provides connection ports between the data acquisition board's analog outputs, analog inputs, and digital input/output pins and the interface electronics. It also routes power to the MEMS gyroscopes and divides the battery voltages so that they can be monitored via the analog input ports.

Circuit diagrams and further explanation of the interface electronics can be found in Appendix C.

2.3 Spacecraft Attitude Dynamics Simulator Software

A suite of software on the PC/104 computer supports the operation of the Spacecraft Attitude Dynamics Simulator by performing tasks ranging from remote control of the on-board computer to generating control commands.

The PC/104 uses Windows XP as the operating system. An ad-hoc wireless network between the PC/104 computer and a ground computer allows the ground computer to take control of the PC/104 via Windows Remote

Desktop. It should be noted, however, that all processing is performed on the PC/104 computer.

Matlab and Simulink are used as controls software. Simulink models are created to gather data, interpret the data, compute control commands, and send the control commands. Matlab functions and scripts support the Simulink models. An open source Simulink block called Simulink Execution Control is used to force Simulink models to run in real time. This block was created by Roger Aarenstrup and is freely available for personal, educational, or professional use.

A suite of Matlab functions was created by Kinnett to allow Matlab and Simulink access to the PC/104 data acquisition device. The Matlab functions eagleAin, eagleAout, and eagleDout read the analog input pins, write to analog output pins, and write to digital output pins respectively.

Chapter III: SADS Dynamics and Torque Control Equation Section 3

In order to create a computer simulation of the SADS as well as to design and test control algorithms, it is important to know the SADS dynamics. The SADS Dynamics are derived by Healy by treating the PRWP as a rigid body. An overview of Healy's derivation is presented here. In addition, the way in which the reaction wheels simulate the existence of a desired external torque will be derived. Next, quaternion kinematics will be briefly presented. Finally, reaction wheel control will be discussed.

3.1 PRWP Geometry

The PRWP has four reaction wheels in a pyramid configuration. Figure 3.1 shows the reference frame configuration which will be referred to throughout this thesis.

Figure 3.1. PRWP Reference Frames⁴

The X, Y , and Z axes and their $\vec{I}, \vec{J}, \vec{K}$ unit vector counterparts represent a fixed inertial reference frame with the origin, O , located at the center of rotation of the PRWP. The x, y , and z axes and their $\vec{i}, \vec{j}, \vec{k}$ unit vector counterparts are fixed to the body frame of the PRWP with the origin at O . The vector $\vec{\omega}$ is the angular rotation rate of the PRWP in the body frame and is defined as $\begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$. The center of mass, CM, of the PRWP is located at $\vec{r}$ from the origin in the body frame.

Figure 3.2 shows the geometry of the reaction wheel pyramid in more detail.

Figure 3.2. Reaction Wheel Geometry⁴

The reaction wheels are aligned with the $+x, +y, -x$, and $-y$ PRWP body axes, but inclined by an angle β , which is equal to 28.3° ³. The reaction wheels' axes of rotation are aligned with the $\vec{i}_1, \vec{i}_2, \vec{i}_3$, and $\vec{i}_4$ unit vectors. These unit vectors define the wheel frames. The state of the reaction wheels can be defined by the

angular velocity and angular acceleration of each wheel, or

$[\omega_{W1} \ \omega_{W2} \ \omega_{W3} \ \omega_{W4}]^T$ and $[\dot{\omega}_{W1} \ \dot{\omega}_{W2} \ \dot{\omega}_{W3} \ \dot{\omega}_{W4}]^T$ respectively.

It will be necessary to transform from the body frame to the inertial frame and from the wheel frames to the body frame via coordinate transformations.

Healy defines

$$\underline{R}_E = \begin{bmatrix} C\theta_2 S\theta_3 & C\theta_2 S\theta_3 & -S\theta_2 \\ S\theta_1 S\theta_2 C\theta_3 - C\theta_1 S\theta_3 & S\theta_1 S\theta_2 S\theta_3 + C\theta_1 C\theta_3 & S\theta_1 C\theta_2 \\ C\theta_1 S\theta_2 C\theta_3 + S\theta_1 S\theta_3 & C\theta_1 S\theta_2 S\theta_3 - S\theta_1 C\theta_3 & C\theta_1 C\theta_2 \end{bmatrix} \quad (3.1)$$

where S and C represent the sine and cosine functions respectively, as the transformation from the body frame to the inertial frame⁴. The angles θ_1, θ_2 , and θ_3 are the Euler rotation angles of the body frame from the X, Y , and Z axes respectively. Healy also defines

$$R_{W1} = \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix}, \quad R_{W2} = \begin{bmatrix} 0 & 1 & 0 \\ \cos \beta & 0 & -\sin \beta \\ \sin \beta & 0 & \cos \beta \end{bmatrix} \quad (3.2), (3.3)$$

$$R_{W3} = \begin{bmatrix} -\cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix}, \quad R_{W4} = \begin{bmatrix} 0 & 1 & 0 \\ -\cos \beta & 0 & \sin \beta \\ \sin \beta & 0 & \cos \beta \end{bmatrix} \quad (3.4), (3.5)$$

as the transformations from the wheel frames to the body frame⁴. The next section describes the PRWP dynamics and how torque commands translate into desired wheel dynamics.

3.2 Torque Generation

The equations of motion of the PRWP are derived by Healy as

$$\vec{r} \times \underline{R}_E \cdot m\vec{g} = \underline{I} \times \dot{\vec{\omega}} + \vec{\omega} \times \underline{I} \vec{\omega} + \sum_{l=1}^4 \left(R_{wl} \cdot \begin{bmatrix} I_{11}^{wl} \cdot \dot{\omega}_{wl} \\ 0 \\ 0 \end{bmatrix} + \vec{\omega} \times R_{wl} \cdot \begin{bmatrix} I_{11}^{wl} \cdot \omega_{wl} \\ 0 \\ 0 \end{bmatrix} \right) \quad (3.6)$$

where $\underline{I}$ is the PRWP's inertia tensor and I_{11}^{wl} is the inertia of the l 'th wheel about the spin axis of that wheel⁴. If the PRWP is balanced properly so that $\vec{r}$ is $[0 \ 0 \ 0]^T$, Eqn. 3.6 can be rearranged as

$$-\sum_{l=1}^4 \left(R_{wl} \cdot \begin{bmatrix} I_{11}^{wl} \cdot \dot{\omega}_{wl} \\ 0 \\ 0 \end{bmatrix} + \vec{\omega} \times R_{wl} \cdot \begin{bmatrix} I_{11}^{wl} \cdot \omega_{wl} \\ 0 \\ 0 \end{bmatrix} \right) = \underline{I} \times \dot{\vec{\omega}} + \vec{\omega} \times \underline{I} \vec{\omega} \quad (3.7)$$

so that the reaction wheels induce an angular momentum of the PRWP equal to the left hand side of Eqn. 3.7. For the wheels to produce the commanded torque, the wheel states, ω_{wl} and $\dot{\omega}_{wl}$, must combine to make the left hand side of Eqn. 3.7 equal to the commanded torque. The commanded input torque is defined as

$$u = \vec{T} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

where T_x , T_y , and T_z are the torques in the PRWP's body frame, so that

$$\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = -\sum_{l=1}^4 \left(R_{wl} \cdot \begin{bmatrix} I_{11}^{wl} \cdot \dot{\omega}_{wl} \\ 0 \\ 0 \end{bmatrix} + \vec{\omega} \times R_{wl} \cdot \begin{bmatrix} I_{11}^{wl} \cdot \omega_{wl} \\ 0 \\ 0 \end{bmatrix} \right) \quad (3.9)$$

The angular velocity of the PRWP, $\vec{\omega}$, and the angular velocity about the spin axis of each reaction wheel, ω_{wl} , are directly measured from the on-board electronics. Therefore, the “gyroscopic” torque, or the second term on the right

hand side of Eqn. 3.9, is a known quantity and can be subtracted from the commanded torque so that

$$\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + \sum_{l=1}^4 \left(\vec{\omega} \times R_{wl} \cdot \begin{bmatrix} I_{11}^{wl} \cdot \dot{\omega}_{wl} \\ 0 \\ 0 \end{bmatrix} \right) = - \sum_{l=1}^4 \left(R_{wl} \cdot \begin{bmatrix} I_{11}^{wl} \cdot \dot{\omega}_{wl} \\ 0 \\ 0 \end{bmatrix} \right) \quad (3.10)$$

Now, define the left hand side of Eqn. 3.10 as $\begin{bmatrix} \hat{T}_x & \hat{T}_y & \hat{T}_z \end{bmatrix}^T$, and

$\begin{bmatrix} -I_{11}^{w1} \cdot \dot{\omega}_{w1} & -I_{11}^{w2} \cdot \dot{\omega}_{w2} & -I_{11}^{w3} \cdot \dot{\omega}_{w2} & -I_{11}^{w4} \cdot \dot{\omega}_{w2} \end{bmatrix}^T$ as $\begin{bmatrix} T_1 & T_2 & T_3 & T_4 \end{bmatrix}^T$. The right

hand side of Eqn. 3.10 can be expanded from Eqns. 3.2, 3.3, 3.4 and 3.5 so that

$$\begin{bmatrix} \hat{T}_x \\ \hat{T}_y \\ \hat{T}_z \end{bmatrix} = \begin{bmatrix} \cos \beta & 0 & -\cos \beta & 0 \\ 0 & \cos \beta & 0 & -\cos \beta \\ \sin \beta & \sin \beta & \sin \beta & \sin \beta \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \quad (3.11)$$

The reaction wheel torques are desired, but the matrix relating the desired torques to the reaction wheel torques is not square and therefore non-invertible.

Sidi²⁰ describes that one possible way to distribute the torques between the four

reaction wheels is to define $\begin{bmatrix} \hat{T}_x / \cos \beta & \hat{T}_y / \cos \beta & \hat{T}_z / \sin \beta \end{bmatrix}^T$ as $\begin{bmatrix} \hat{T}'_x & \hat{T}'_y & \hat{T}'_z \end{bmatrix}^T$ so

that

$$\begin{bmatrix} \hat{T}'_x \\ \hat{T}'_y \\ \hat{T}'_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = A_w \cdot \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \quad (3.12)$$

The matrix relating the desired torques to the reaction wheel torques, A_w , is still

not square, but by taking the right pseudo-inverse of A_w as

$$A_{wR} = A_w^T \cdot (A_w \cdot A_w^T)^{-1} = \frac{1}{2} \cdot \begin{bmatrix} 1 & 0 & 1/2 \\ 0 & 1 & 1/2 \\ -1 & 0 & 1/2 \\ 0 & -1 & 1/2 \end{bmatrix} \quad (3.13)$$

Eqn. 3.12 can be rewritten as

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} 1 & 0 & 1/2 \\ 0 & 1 & 1/2 \\ -1 & 0 & 1/2 \\ 0 & -1 & 1/2 \end{bmatrix} \cdot \begin{bmatrix} \hat{T}'_x \\ \hat{T}'_y \\ \hat{T}'_z \end{bmatrix} \quad (3.14)$$

Sidi²⁰ notes that this torque distribution minimizes the norm of $[T_1 \ T_2 \ T_3 \ T_4]^T$.

The next section describes how the reaction wheels will produce these torques.

3.3 Quaternion Kinematics

Quaternion kinematics will be used extensively, so they are briefly presented here. Quaternion kinematics is advantageous over Euler angle kinematics for spacecraft because quaternion kinematics does not suffer from singularities as Euler angle kinematics does. Quaternion kinematics as defined by Wie²¹ is

$$\dot{\vec{q}} = (q_4 \vec{\omega} - \vec{\omega} \times \vec{q}) \quad (3.15)$$

$$\dot{q}_4 = -\frac{1}{2} \vec{\omega}^T \vec{q} \quad (3.16)$$

where $\vec{q}$ and q_4 are the scalar and vector parts, respectively, of the quaternion

$$q = [\vec{q} \ q_4]^T.$$

For control purposes, it will be necessary to define the error in the orientation of the PRWP. A simple difference between the desired quaternion and the actual quaternion does not sufficiently describe the orientation error

because it is not equivalent to describing the Euler vector and the rotation about that vector that is needed to arrive at the commanded orientation. Instead, Full State Feedback control uses the error quaternion, q_e , defined as

$$\begin{bmatrix} \vec{q}_e \\ q_{4e} \end{bmatrix} = \begin{bmatrix} q_{4c} & q_{3c} & -q_{2c} & -q_{1c} \\ -q_{3c} & q_{4c} & q_{1c} & -q_{2c} \\ q_{2c} & -q_{1c} & q_{4c} & -q_{3c} \\ q_{1c} & q_{2c} & q_{3c} & q_{4c} \end{bmatrix} \cdot \begin{bmatrix} \vec{q}_a \\ q_{a4} \end{bmatrix} \quad (3.17)$$

where the subscript “c” represents the commanded quaternion, and the subscript “a” represents the actual quaternion.

3.4 Reaction Wheels

The reaction wheels are designed to track wheel speed commands via a Proportional, Integral, Derivative (PID) controller. Therefore, the desired torque from each reaction wheel must be converted into a desired wheel speed. The desired torques from each wheel, $[T_1 \ T_2 \ T_3 \ T_4]^T$ were derived in the previous section. Recall that these torques were defined as

$$\begin{bmatrix} -I_{11}^{W1} \cdot \dot{\omega}_{W1} & -I_{11}^{W2} \cdot \dot{\omega}_{W2} & -I_{11}^{W3} \cdot \dot{\omega}_{W2} & -I_{11}^{W4} \cdot \dot{\omega}_{W2} \end{bmatrix}^T \text{ so that the desired wheel speeds}$$

for each reaction wheel can be written as

$$\omega_{Wl}(t) = -\frac{1}{I_{11}^{Wl}} \int T_l(t) dt + \omega_{Wl}^i \quad (3.18)$$

where ω_{Wl}^i is the initial angular velocity of the l 'th wheel. The moment of inertia of each wheel, I_{11}^{Wl} , is estimated to be $.0005355 \text{ kg} \cdot \text{m}^2$. Prior to performing a simulation using the PRWP, the reaction wheels are accelerated to an initial angular velocity. Biasing the reaction wheels this way helps minimize the number

of times the motor needs to change direction. It can be difficult to ensure a smooth transition from one direction to another, which may result in undesired dynamics. Also, an angular momentum bias can help keep the PRWP stable by introducing the “gyroscopic” torque terms to the PWRP dynamics.

The wheel speed measurements that are read by the A/D converters from the frequency to voltage conversion circuit are inherently noisy. Figure 3.3 shows angular velocity measurements at a number of constant PWM duty cycles for one of the reaction wheels.

Figure 3.3 Wheel Speed Measurement Noise

Measurements were taken for ten seconds after the reaction wheel was close to steady state speeds. It is evident that the reaction wheels are not actually experiencing spikes in the angular velocity in excess of 30 rad/s during a single

.02 second time step. The proportional and derivative components of the PID controller will overcompensate for these spikes. Therefore, it is necessary to filter the wheel speed measurements.

A software implemented Kalman filter was selected to clean up the noisy measurements. Figure 3.4 shows both the unfiltered and Kalman filtered wheel speeds for one reaction wheel.

Figure 3.4 Wheel Speed Kalman Filter

The Kalman filtered reaction wheel speed measurements are shown on top while the unfiltered measurements are shown on bottom. Inspection demonstrates that short spikes are significantly suppressed. This helps the PID controller track wheel speed commands accurately without overcompensating for measurement

spikes. In addition to suppressing spikes, the Kalman filter preserves upward and downward trends in the wheel speed measurements.

A PID controller is used to control the wheel speeds because it is simple to implement and easy to tune. PID controllers are perhaps the most widely used controllers and find use in everything from radio controlled airplanes to industrial processes. Details about PID controllers and gain tuning can be found in Nise²².

If the wheel speed error of the l 'th wheel is defined as

$$e_l = \omega_{wl}^d - \omega_{wl}^m \quad (3.19)$$

where ω_{wl}^d and ω_{wl}^m are the desired and Kalman filtered angular velocities of the l 'th wheel respectively, then the output of the PID controller to the l 'th wheel is

$$u_l = K_p e_l + K_I \int e_l dt + K_D \frac{de_l}{dt} \quad (3.20)$$

where K_p , K_I , and K_D are the constant proportional, integral, and derivative gains respectively. The reaction wheel responses to initial guesses of K_p , K_I , and K_D were examined and the gains were tuned manually until a suitable controller was found. The output of the PID controller is fed into a logic block in Simulink that determines what voltages to output to the PWM generation circuit and motor drivers to best emulate the PID controller's desired input to the reaction wheel dynamic system.

The reaction wheels are limited by maximum and minimum wheel speeds as well as maximum acceleration and deceleration. Each reaction wheel is nominally capable of speeds of up to 6000 revolutions per minute (RPM) (628 rad/s) in both directions. However, for safety purposes, the commanded wheel

speed is limited between 500 and 5000 RPM (52 rad/s to 520 rad/s). Currently, the reaction wheels are not allowed to change their direction of rotation. This is because the direction determination circuit does not work properly. Instead, the wheels always rotate in a single direction at speeds between the upper and lower limits. If the desired wheel speed is higher than the limit, the commanded wheel speed is equal to the upper limit. If the desired wheel speed is lower than the lower limit, the commanded wheel speed is equal to the lower limit. The acceleration and braking of the reaction wheels are limited by the PWM duty cycle. The reaction wheel logic block in Simulink limits the voltage that goes into the PWM generation circuit, thereby limiting the PWM duty cycle that is produced. The reason for this is that the motor driver circuit is susceptible to overheating when the reaction wheels accelerate or brake too much. The maximum acceleration and deceleration of the reaction wheels is approximately 60 rad/s^2 . This results in a maximum torque per reaction wheel of approximately .032 Nm.

Chapter IV: PRWP Attitude Control Equation Section 4

Attitude control of the PRWP will be performed by Full State Feedback (FSFB), NDMRAC, and AOF. Computer simulations of all three control laws will be compared to actual simulations using the SADS. Three cases of simulations will be run. The nominal case is an attitude maneuver simulation when the mass properties of the PRWP are well known. The second case is an attitude maneuver when the mass properties are roughly estimated. For the final case, the PWRP will attempt to hold its orientation during simulated actuator damage.

4.1 Full State Feedback Control Overview

Full State Feedback control applied to the rigid body equations of motion generates a commanded torque based on the angular velocity and error quaternion of the form

$$u = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = -K \cdot \vec{q}_e - C\vec{\omega} \quad (4.1)$$

where K and C are the 3x3 FSFB control gains. Asymptotic stability has been proven for certain values of K and C by Mittelsteadt and Wie^{3,21}. This thesis will make use of the method that Mittelsteadt presents to calculate the FSFB gains.

Mittelsteadt³ defines the FSFB gains as

$$K = \begin{bmatrix} 2I_{xx}\omega_n^2 & 0 & 0 \\ 0 & 2I_{yy}\omega_n^2 & 0 \\ 0 & 0 & 2I_{zz}\omega_n^2 \end{bmatrix} \quad (4.2)$$

$$C = \begin{bmatrix} 2I_{xx}\xi\omega_n & 0 & 0 \\ 0 & 2I_{yy}\xi\omega_n & 0 \\ 0 & 0 & 2I_{zz}\xi\omega_n \end{bmatrix} \quad (4.3)$$

where ω_n and ξ are the specified natural frequency and damping ratio, respectively, of the linearized quaternion kinematics equation,

$$\frac{d^2 \vec{q}_e}{dt^2} + C \underline{I}^{-1} \frac{d \vec{q}_e}{dt} + \frac{1}{2} K \underline{I}^{-1} \vec{q}_e = 0 \quad (4.4)$$

The natural frequency and damping ratio can be selected based on second order differential equation response metrics, such as settling time and percent overshoot.

4.2 NDMRAC and AOF Overview

Cal Poly graduate students Torres and Patel developed NDMRAC and AOF respectively. An overview of both control laws will be presented here. NDMRAC and AOF operate by changing adaptive control gains based on the system outputs' deviation from a reference model. AOF is a simplified version of NDMRAC. Figure 4.1 shows the block diagram of a generic direct adaptive model reference controller.

Figure 4.1 Generic Direct Adaptive Model Reference Controller

For NDMRAC and AOF, the plant can be nonlinear and receives M inputs, $\underline{u}_p$, and generates M outputs, $\underline{y}_p$, but can have more than M states. The reference model is user-defined and has N states, $\underline{x}_m$. It generates M outputs, $\underline{y}_m$, from M inputs, $\underline{u}_m$. The error between the reference model outputs and the plant outputs is defined as

$$\underline{e}_y = \underline{y}_p - \underline{y}_m \quad (4.5)$$

The system input from the NDMRAC control law is

$$\underline{u}_p = G_e \underline{e}_y + S_{21} \underline{x}_m + S_{22} \underline{u}_m \quad (4.6)$$

and the system input from the AOF control law is

$$\underline{u}_p = G_e \underline{e}_y \quad (4.7)$$

where G_e , S_{21} , and S_{22} are the adaptive output, state, and input gains, respectively.

The rates of change of the adaptive gains of the controllers are defined as

$$\dot{S}_{21} = -\underline{e}_y \underline{x}_m^T H_1 \quad (4.8)$$

$$\dot{S}_{22} = -\underline{e}_y \underline{u}_m^T H_2 \quad (4.9)$$

$$\dot{G}_e = -\underline{e}_y \underline{e}_y^T H_3 \quad (4.10)$$

where H_1 , H_2 , and H_3 are constant positive definite matrices known as the adaptive parameters. For a proof of stability and more information about NDMRAC, the reader is referred to Torres¹⁷. For a proof of stability and more information about AOF, the reader is referred to Patel¹⁸.

Both NDMRAC and AOF impose limitations on the system. The stability proofs for both NDMRAC and AOF assume that the system is square, minimum

phase, and strictly positive real. Torres and Patel apply NDMRAC and AOF to rigid body equations of motion, indicating that the PRWP plant is indeed minimum phase and strictly positive real. However, there are three inputs to the PRWP (the three body torques) and six outputs (the vector part of the quaternion and three body rates), so the system is not square. To fix this, the first three inputs are added to the second three inputs.

4.3 Test Cases

The FSFB, NDMRAC, and AOF controllers will each be tested under three conditions. For the first test case, each controller will be designed based on the accurate PRWP mass properties determined from the system identification algorithm. The system identification process determined that the actual inertia tensor of the PRWP, $\underline{I}_a$ is

$$\underline{I}_a = \begin{bmatrix} .76 & -.035 & -.036 \\ -.035 & .70 & -.017 \\ -.036 & -.017 & .59 \end{bmatrix} kg \cdot m^2 \quad (4.11)$$

These tests will be used as a control for comparing nominal performance of each controller.

The second test case is intended to emulate a “Plug-and-Play” satellite scenario. For the second test case, each controller will be designed based on inaccurate mass property data generated by a simple solid model of the PRWP generated using Unigraphics NX 6. Figure 4.2 shows the solid model of the PRWP.

Figure 4.2 Solid Model of the PRWP

The inertia tensor generated by the Solid Works model, $\underline{I}_{est}$, is

$$\underline{I}_{est} = \begin{bmatrix} .58 & .0265 & -.0023 \\ .0265 & .58 & .0062 \\ -.0023 & .0062 & .58 \end{bmatrix} kg \cdot m^2 \quad (4.12)$$

The percent error of the estimated inertia tensor is

$$\frac{\underline{I}_{est} - \underline{I}_a}{\underline{I}_a} \times 100\% = \begin{bmatrix} -25 & -176 & -93 \\ -176 & -17 & -137 \\ -93 & -137 & -1.5 \end{bmatrix} \% \quad (4.13)$$

Each controller will be designed based on the estimated inertia matrix.

The third test will emulate actuator damage by degrading the performance of the fourth reaction wheel. To achieve this, the wheel speed commands to the fourth reaction wheel will be altered to emulate a slower response time and more overshoot. The nominal commanded wheel speeds will be passed through the transfer function

$$\frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (4.14)$$

where ω_n is the natural frequency and ξ is the damping ratio of the simulated degraded reaction wheel dynamics. The natural frequency, ω_n , was chosen as 1 and the damping ratio, ξ , was chosen as 1.5. Figure 4.3 compares nominal reaction wheel performance to the performance of the fourth reaction wheel during the third test.

Figure 4.3 Nominal and Degraded Reaction Wheel Speeds

Each controller will be designed based on the actual inertia matrix. The adaptive parameters and initial gains of the NDMRAC and AOF controllers will be the same as those used in the second test.

Computer and SADS simulations will be performed for all three controllers and each of the three test cases. The performance of each controller will be compared.

4.4 Implementation of FSFB

As discussed in Section 4.2, the FSFB gains are selected based on desired performance of the linearized quaternion kinematics. The controller was designed so that Eqn. 4.7 has a 1% settling time of 20 seconds and a damping ratio, ξ , of .75. The natural frequency, ω_n , of a second order system is

$$\omega_n = \frac{-\ln(P)}{\xi T_s} \quad (4.15)$$

where T_s is the settling time and P is the percentage settled. The K and C gains for the FSFB controller for the first and second test cases

are $K = \begin{bmatrix} .145 & 0 & 0 \\ 0 & .132 & 0 \\ 0 & 0 & .111 \end{bmatrix}$ and $C = \begin{bmatrix} .354 & 0 & 0 \\ 0 & .322 & 0 \\ 0 & 0 & .271 \end{bmatrix}$. The K and C gains for the

FSFB controller for the second test case are $K = \begin{bmatrix} .108 & 0 & 0 \\ 0 & .108 & 0 \\ 0 & 0 & .108 \end{bmatrix}$ and

$$C = \begin{bmatrix} .265 & 0 & 0 \\ 0 & .265 & 0 \\ 0 & 0 & .265 \end{bmatrix}.$$

4.5 Implementation of NDMRAC and AOF

The reference model for the NDMRAC and AOF controllers is user-defined. The reference model was chosen as an idealized FSFB controller

applied to the PRWP equations of motion. The FSFB controller is the same controller as is used in FSFB control. A block diagram of the reference model is shown in Figure 4.4.

Figure 4.4 Reference Model

The PRWP equations of motion presented in Eqn. 3.6 are integrated to simulate the response to the FSFB controller inputs. The input to the reference model is simply the desired quaternion. The reference model states are the vector part of the quaternion, the derivative of the vector part of the quaternion, the body rates, and the derivative of the body rates. The output from the reference model is the vector part of the quaternion and the body rates.

A good method for determining the adaptive parameters and the initial conditions of the adaptive gains analytically based on the plant is unknown. Patel uses an optimization technique to determine the adaptive parameters and initial conditions. In this thesis, however, the adaptive parameters and initial gains for the NDMRAC and AOF controllers were determined recursively. An initial guess of the initial gains and adaptive parameters was made and the computer simulated response to an attitude maneuver was made. During the course of the

maneuver, the gains adapt to a better value. These gains are then used as the initial gain values and the computer simulation is performed again. This process is repeated until the initial gains do not change much over the course of the maneuver. Next, the adaptive parameters were roughly chosen so that the system would be robust to changes in the inertia tensor. The final values chosen for the initial gains and adaptive parameters can be found in the m-files in Appendix A.

4.6 Computer Model of Simulator

The computer model of the simulator operates by integrating the PRWP equations of motion defined by Eqn 3.6 to produce simulated plant outputs, which in turn are then used by a controller to produce simulated plant inputs. Additionally, measurement noise and reaction wheel limitations and dynamics are incorporated into the simulation. The block diagram of the computer model of the simulator is shown in Figure 4.5.

Figure 4.5 Computer Model of Simulator

It is assumed that the center of gravity of the PRWP is co-located with the pivot point so that there are no external torques on the PRWP. It will be demonstrated in Chapter V that this model of the simulator is sufficiently accurate for nominal operation of the SADS, but breaks down when large torques are commanded by the controller.

Chapter V: Simulation Results

Each controller was simulated using the computer simulation and the SADS. The computer simulations indicate that the NDMRAC and AOF controllers are capable of providing attitude control at least as well as the FSFB controller during the first and second test cases. The computer simulations indicate that all three controllers are capable of eventually arriving at the commanded quaternion, but exhibit different transient behavior in doing so. The SADS simulations show that all three controllers are indeed capable of providing attitude control under nominal conditions. The third test case shows that all three controllers are still capable of performing the desired maneuver in spite of the simulated actuator damage.

5.1 Desired Quaternion

The desired quaternion trajectory is the same for all test cases. For the first twenty seconds of the simulation, the PRWP is commanded to remain in the “home” attitude at the quaternion $q = [0 \ 0 \ 0 \ 1]^T$. From 20 seconds to the end of simulation at 70 seconds, the commanded quaternion is changed to $q = [.145 \ .111 \ .145 \ .972]^T$.

5.2 Computer Simulation Results

The first test case serves as a baseline for comparing the three controllers. Each controller was designed based on the mass properties gathered from the system identification algorithm. Figure 5.1 shows a computer simulation

of the trajectory of the vector part of the quaternion, $\vec{q} = [q_1 \ q_2 \ q_3]^T$, for the first test case.

Figure 5.1 Computer Simulation of First Test Case

Each controller performs equally well at arriving at the desired quaternion.

Overall performance of each controller is very comparable, but there are slight differences in the trajectory that each controller takes to perform the maneuver.

This is because the initial gains that were chosen for the adaptive controllers are not ideal gains. The initial gains are shown to perform well, but do not necessarily provide ideal tracking of the reference model.

The second test case demonstrates that the performance of all three controllers does not decrease significantly when the controllers are designed based on the estimated inertia matrix. Figure 5.2 shows the computer simulated quaternion vector trajectories of each controller for the second test case.

Figure 5.2 Computer Simulation of Second Test Case

Once again, each controller arrives at the desired quaternion as expected. The settling time of each controller is approximately the same. The computer simulation of the second test case indicates that the actual performance of each controller should not be significantly different between the first and second test cases.

The third test case demonstrates the differences in how each controller responds to the reaction wheel with degraded performance. Figure 5.3 shows the computer simulated quaternion vector trajectories of each controller for the third test case.

Figure 5.3 Computer Simulation of Third Test Case

All three controllers do a reasonably good job of arriving at the desired quaternion despite the degraded reaction wheel performance. The AOF controller settles at the desired quaternion faster than the NDMRAC and FSFB controllers. All three controllers exhibit damped oscillations as the quaternion trajectory approaches the desired quaternion. The FSFB controller's oscillations

are of a lower frequency than the NDMRAC and AOF controllers, indicating that the NDMRAC and AOF controllers are coping with the degraded reaction wheel performance by adapting into higher gain systems. This indication is confirmed by examining the commanded body torques. Figure 5.4 shows the body torque commands generated by each controller during the third test case. The body torques commanded by the NDMRAC and AOF controllers are of a higher amplitude than the torques commanded by the FSFB controller.

Figure 5.4 Computer Simulated Body Torques for Third Test Case

The NDMRAC and AOF controllers produce better transient performance during the maneuver at the expense of actuator effort.

5.3 SADS Simulation Results

The SADS simulations are performed for each test case and compared to the computer simulations. The quaternion vector trajectories from the SADS simulations of first test case are shown in Figure 5.5.

Figure 5.5 SADS Simulation of First Test Case

The FSFB controller causes the PRWP to oscillate during the first 20 seconds and then converge to a quaternion that is offset from the desired quaternion. The NDMRAC and AOF controllers track the reference model well, but oscillate more than the computer simulations predicted. All three controllers are effective at controlling the attitude of the PRWP. This is consistent with the results from the computer simulations. Figure 5.6 shows the final 40 seconds of the simulation zoomed in so that the oscillation is more visible.

Figure 5.6 Zoom View of SADS First Test Case

The oscillation and FSFB convergence behaviors indicate two things. First, the SADS system exhibits more measurement and actuator noise than was

simulated on the computer. In addition to being noisy, the gyroscopes drift and although the platform may not actually move much, the quaternion measurement is constantly drifting. The controllers are telling the actuators to track towards this drifting quaternion. This explains much of the oscillation that occurs for all three controllers. The second indication is that the center of gravity of the platform is not exactly aligned with the pivot point of the air bearing. The FSFB controller generates a torque command based on the error quaternion and angular rates. When the angular rates are low, the torque is based mostly on the error quaternion. If a small torque due to gravity is always present, the FSFB controller will settle at a quaternion in such a way that the commanded torque from the controller is exactly opposing the gravity torque. It is impossible to place the center of gravity exactly at the pivot point, so this effect will always be present for the FSFB controller. The NDMRAC and AOF controllers adapt themselves to track the reference model, so they do not exhibit this behavior. This indicates that the NDMRAC and AOF controllers are capable of rejecting simple disturbances without the use of dedicated disturbance rejection. Although the gravity torque and gyroscope drift affect the fidelity of the SADS simulations, this first test case can be used as a baseline for comparing the second and third test case simulations.

The quaternion trajectories for the SADS simulations of the second test case are shown in Figure 5.7.

Figure 5.7 SADS Simulation of Second Test Case

The results from the second test case are very similar to the results from the first test case. The oscillation and FSFB convergence offset problems discussed earlier are still present. All three controllers converged to the desired quaternion about as well as they did for the first test case, but because of the oscillation and FSFB convergence offset, it is difficult to make strong comparisons between the two test cases. The results of this test are consistent with the computer simulation results.

The results from the SADS simulations of the third test case are shown in Figure 5.8.

Figure 5.8 SADS Simulation of Third Test Case

Only the FSFB controller converges to a steady quaternion, although it is offset as in the first and second test cases. Other than the divergence from the desired quaternion towards the end of the adaptive controllers' simulations, overall performance of all three controllers is similar to the performance predicted by the computer simulation. All three controllers exhibit oscillations as they approach the desired quaternion. The FSFB controller oscillations are low frequency compared to the NDMRAC and AOF controller oscillations. As expected, the NDMRAC and AOF controllers adapted to the degraded reaction wheel

performance by becoming a higher gain system. This appeared to be effective for approximately 7 seconds, but eventually the NDMRAC and AOF controllers commanded torques that the reaction wheels could not produce. The body torques commanded by these controllers are shown in Figure 5.9.

Figure 5.9 NDMRAC and AOF Third Test Body Torques

At approximately 20 seconds into the simulation, the NDMRAC controller commanded a torque that the reaction wheels could not produce and more error is introduced into the system. After this happened, the controller adapts by moving towards a higher gain system. Eventually, torques in excess of 1000 N-m were commanded by the NDMRAC controller. Obviously, the reaction wheels

can not produce these torques. A similar phenomenon occurs with the AOF controller. Part of the proofs of stability of the NDMRAC and AOF controllers is that the plant is minimum phase. The reaction wheels have acceleration and deceleration limits, saturation limits, and small phase delay between the commanded speed and actual speed. It appears that a combination of some or all of these effects may violate the minimum phase requirement of the NDMRAC and AOF stability proofs. The controller needs to command torques that the reaction wheels can reasonably produce. Since the computer simulation converged, it is evident that the computer model does not accurately represent the reaction wheel dynamics.

As a demonstration that the NDMRAC and AOF controllers can be used to compensate for the degraded reaction wheel dynamics, the third adaptive parameter, H_3 , was changed from $1000 \times I_{6 \times 6}$ to $500 \times I_{6 \times 6}$ and the third test case was performed again. The quaternion trajectories from these SADS simulations are shown in Figure 5.10.

Figure 5.10 NDMRAC and AOF with Decreased Adaptive Parameter

The NDMRAC and AOF controllers both track the reference model fairly well and begin to converge to the desired quaternion. In this case, the torques commanded by the controllers do not make the reaction wheels saturate, nor do they command torques in excess of the reaction wheels' capabilities. It appears that the minimum phase requirement is met in this case. Although the third adaptive parameter is half of what it was before, the SADS simulations still behave very similarly to the computer simulations shown in Figure 5.3. The oscillations produced by the AOF controller are of lower magnitude than the NDMRAC controller. The AOF controller has better performance than the NDMRAC controller in this test case.

Chapter six discusses the importance of the computer and SADS simulation results and the conclusions that can be drawn from them.

Chapter VI: Conclusion

The results of the computer and SADS simulations discussed in Chapter five successfully demonstrate proof of concept of NDMRAC and AOF as spacecraft attitude controllers. The adaptive controllers were shown to have advantages and disadvantages over the FSFB controller. It is clear that adaptive controllers need to be designed carefully in order to avoid unwanted behavior. There are several recommendations for future study that may improve adaptive controller design.

6.1 Computer and SADS Simulation Conclusions

A number of important conclusions can be drawn from the computer and SADS results. There are several differences in the results from the computer simulation and the SADS simulation that are enlightening. First, it became clear that although the PRWP appeared to be balanced well, there is still a torque due to gravity. This torque was sufficient enough to create a noticeable offset in the FSFB controller's steady-state quaternion. This steady-state offset could easily be remedied by the inclusion of an integral gain term in the FSFB controller. Second, the SADS actuator and measurement noise was larger and played a more significant role in the control of the PRWP than the computer simulation indicated. The design of the NDMRAC and AOF controllers was based on the computer model. The NDMRAC and AOF computer simulations of the third test case converged but the SADS simulations did not. This indicates that good knowledge of the kinds of measurement and actuator errors that may be present

in the plant is necessary to safely design these adaptive controllers, although the actual errors may be unknown but within certain bounds.

The results of the second test indicate two things. First, the measurement and actuator noise of the SADS needs to be reduced in order to make firm conclusions about the effectiveness of the adaptive controllers at compensating for differences in the design inertia tensor and actual inertia tensor. The inclusion of the LN-200 gyroscope will significantly improve knowledge of the SADS attitude. The FSFB controller performed well during the second test. In a “Plug-and-Play” satellite scenario, if the inertia tensor is the only unknown system parameter, a simple FSFB controller may still be sufficient to provide tracking and pointing capabilities. However, for systems with more uncertainties or more demanding pointing accuracy requirements, adaptive control may be a better option.

The results of the third test case indicate that the FSFB controller is capable of recovering from an actuator failure as in the third test case. The FSFB controller experienced reduced performance, but eventually started to converge to the desired quaternion. The NDMRAC and AOF controllers caused the SADS to go unstable. However, in the test case where the adaptive parameter was halved, the NDMRAC and AOF controllers performed well. The results of the NDMRAC and AOF tests with the third adaptive parameter reduced show that the transient behavior of these adaptive controllers is sensitive to the adaptive parameters. It may be possible that there is an upper bound on what the adaptive parameters can be to guarantee asymptotic stability based on the actuator

capabilities. In general, questions like this need to be addressed before adaptive controllers are used more often in high-cost engineering projects.

Overall, the measurement limitations of the SADS prevented in depth comparison between the three controllers. Nonetheless, the tests provide proof of concept of the NDMRAC and AOF controllers applied to the attitude control of the SADS.

6.2 Future Recommendations

Based on the conclusions stated earlier, it is clear that better measurement sensors are needed on the SADS. The LN-200 gyroscope will greatly improve knowledge of the PRWP states. This will assist in achieving more accurate control. Also, knowledge of the location of center of mass of the PRWP may improve controller performance. Further, there is always room for improved reaction wheel control by improving wheel rate sensing and command as well as the ability to rotate both clockwise and counterclockwise and to transition smoothly between these directions. These changes could allow the SADS to obtain very accurate data that can be used to make firm conclusions about controller performance.

As stated earlier, it may be possible to place bounds on the adaptive parameters so that stability of realistic plants can be proven. The SADS would be a good testbed for verifying analytical methods of determining these bounds. Additionally, the adaptive controller may be more effective if the error quaternion is used rather than the difference in the desired and actual quaternion. Finally, a

robust method of determining initial conditions of the adaptive gains would be helpful in the design of these adaptive controllers.

List of References:

- ¹Cookson, S., "LabVIEW Stabilization Program: Inertial/Reaction Wheel Platform," Senior Project, Aerospace Engineering Dept., California Polytechnic State Univ., San Luis Obispo, CA, 2005.
- ²Martin, J., Nash, A., and Quinto, A., "Reaction Wheel Platform (Satellite Simulator Structure)," Senior Project, Mechanical Engineering Dept., California Polytechnic State Univ., San Luis Obispo, CA, 2006.
- ³Mittlesteadt, C., "Results on the Development of a Four-Wheel Pyramidal Reaction Wheel Platform," Master's Thesis, Aerospace Engineering Dept., California Polytechnic State Univ., San Luis Obispo, CA, 2006.
- ⁴Healy, P., "Mass Property System Identification of a Spacecraft Simulator," Master's Thesis, Aerospace Engineering Dept., California Polytechnic State Univ., San Luis Obispo, CA, 2006.
- ⁵Saile, C., "Center of Mass Fine Tuning System for the Cal Poly Spacecraft Simulator," Senior Project, Aerospace Engineering Dept., California Polytechnic State Univ., San Luis Obispo, CA, 2007.
- ⁶Liu, A., Luo, V., and Tam, W., "Bluetooth Network Data Acquisition from Four-Wheel Pyramidal Reaction Wheel Platform," Senior Project, Computer Engineering Dept. and Electrical Engineering Dept., California Polytechnic State Univ., San Luis Obispo, CA, 2007.
- ⁷Logan, J., "Control and Sensor Development on a Four-Wheel Pyramidal Reaction Wheel Platform," Master's Thesis, Aerospace Engineering Dept., California Polytechnic State Univ., San Luis Obispo, CA, 2008.
- ⁸Silva, S., "Applied System Identification for a Four Wheel Reaction Wheel Platform," Master's Thesis, Aerospace Engineering Dept., California Polytechnic State Univ., San Luis Obispo, CA, (to be published).
- ⁹Nguyen, N., Krishnakumar, K., Kaneshige, J., and Nespeca, P., "Dynamics and Adaptive Control for Stability Recovery of Damaged Asymmetric Aircraft," *AIAA Guidance, Navigation, and Control Exhibit*, Keystone, CO, Aug. 21-24, 2006.
- ¹⁰Liu, Y., and Tao, G., "Model-Based Direct Adaptive Actuator Failure Compensation Techniques for Aircraft Flight Control," *AIAA Guidance, Navigation, and Control Exhibit*, Keystone, CO, Aug. 21-24, 2006.

¹¹Shin, Y., Calise, A., and Johnson, M., "Adaptive Control of Advanced Fighter Aircraft in Nonlinear Flight Regimes," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, Sept. - Oct. 2008, pp. 1464-1477.

¹²Lee, K., Nambisan, P., and Singh, S., "Adaptive Variable Structure Control of Aircraft with an Unknown High-Frequency Gain Matrix," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 1, Jan. - Feb. 2008, pp. 194-203.

¹³Shageer, H., and Tao, G., "Modeling and Adaptive Control of Spacecraft with Fuel Slosh: Overview and Case Studies," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Hilton Head, SC, Aug. 20-23, 2007.

¹⁴Kharisov, E., Gregory, I., Cao, C., and Hovakimyan, N., " $\mathcal{L}_1$ Adaptive Control Law for Flexible Space Launch Vehicle and Proposed Plan for Flight Test Validation," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Honolulu, HI, Aug. 18-21, 2008.

¹⁵Scarritt, S., "Nonlinear Model Reference Adaptive Controller for Satellite Attitude Tracking," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Honolulu, HI, Aug. 18-21, 2008.

¹⁶Harvey, S., and Balas, M., "Direct Model Reference Adaptive Attitude Control of a PNP Satellite with Unknown Dynamics," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Hilton Head, SC, Aug. 20-23, 2007.

¹⁷Torres, S., "Nonlinear Adaptive Control for Spacecraft Rigid Body Equations of Motion," Master's Thesis, Aerospace Engineering Dept., California Polytechnic State Univ., San Luis Obispo, CA, 2006.

¹⁸Patel, R., "Adaptive Output Feedback Control as Applied to the Rigid Body Equations of Motion," Master's Thesis, Aerospace Engineering Dept., California Polytechnic State Univ., San Luis Obispo, CA, 2007.

¹⁹Kinnett, R., "System Integration and Control of a Low-Cost Spacecraft Attitude Dynamics Simulator," Master's Thesis, Aerospace Engineering Dept., California Polytechnic State Univ., San Luis Obispo, CA, (to be published).

²⁰Sidi, M., "Attitude Dynamics and Kinematics," *Spacecraft Dynamics and Control*, Cambridge University Press, New York, 1997.

²¹Wie, B., *Space Vehicle Dynamics and Control*, AIAA, Reston, VA, 1998.

²²Nise, N., *Control Systems Engineering*, 4th Ed., Wiley, Hoboken, NJ, 2003.

Appendix A: Matlab Code

A.1 SADS Simulation Tests Setup m-file

```
1 % TestsSetup.m
2 % Matt Downs
3 % This m-file sets up the SADS simulation
4
5 clear; clc;
6
7 %% SENSOR SETUP
8
9 dt = 0.02; %time (s) per iteration
10 n = 4; %select motor # (1-4)
11 m=2;
12
13 load 'MotorSetup.mat';
14 load 'rwcal.mat';
15 v2f = volts2freq(n);
16
17 cmdpin=Motor(n).cmdpin;
18 spdpin=Motor(n).spdpin;
19 dxnpin=Motor(n).dxnpin;
20 dxnpins=Motor(n).dxnpins;
21
22 %WHEEL CONTROLLER GAINS:
23 Kp = 0.03;
24 Ki = 0.0001;
25 Kd = 0.001;
26
27 biasspeed = [3800,2200,2200,3800] % Wheel Bias Speed (RPM)
28 %eagleDout4(1000019524,[Motor(:).dxnpins],[1 0 1 0 1 0 1 0]); %set
all to ccw
accel mode
29
30 %V0 = eagleAinAvg(1000019524,[Motor(:).spdpin],1000);
31 disp('Calibrating reaction wheels.');
```

```
32 sim('RwOffsetCal');
33 disp('...done.');
```

```
34
35 for n=1:3, mexBeep( 2000, 50); pause(0.05); end;
36 disp('Calibrating Gyros. Keep er steady');
```

```
37 sim('GyroCal');
```

```
38 disp('...done.');
```

```
39
40 %%
41 trial = 'C4_';
42 pname = 'C:\SpaceSim_Software\MattsTests\SavedData\';
43 %% Which Case Are You Doing and With What Controller?
44 % Which cases do you want to run?
45 Cases = [1];
46 % Case I: Nominal Case. Good RW's and Known Inertia Tensor
47 % Case II: Good RW's and Estimated Inertia Tensor
48 % Case III: RW 4 is bad, Known Inertia Tensor
49 % Which controllers do you want to use?
```

```

50 Controllers = [3];
51 % Controller I: Full State Feedback
52 % Controller II: NDMRAC
53 % Controller III: AOF
54 % Do you want to save the data? Y = 1, N = 0;
55 Save = 1;
56 % Simulation Time, seconds
57
58 %% Perform Desired Cases
59 Case = Cases(1);
60 Controller = Controllers(1);
61
62 % System Information
63 % PRWP Mass Properties
64 I = [.768,-.0348,-.0355;-.0348,.699,-.0165;-.0355,-.0165,.589]; %
SysID PRWP
Inertia, kg-m^2
65 Iinv = inv(I); % Inverse of inertia tensor
66 Idia = [I(1,1),0,0;0,I(2,2),0;0,0,I(3,3)]; % Linearized PRWP
Inertia, kgm^
2
67 Iinvdia = inv(Idia); % Inverse of linearized inertia tensor
68 Iest = [.575,.0265,-.00234;.0265,.576,.00615;-.00234,.00615,.580]; %
Estimated PRWP Inertia, kg-m^2
69 Iinvest = inv(Iest); % Inverse of estimated inertia tensor
70 Idiaest = [Iest(1,1),0,0;0,Iest(2,2),0;0,0,Iest(3,3)]; % Linearized
PRWP
Estimated Inertia, kg-m^2
71 Iinvdiaest = inv(Idiaest); % Inverse of estimated inertia
linearized
tensor
72 if Case == 2;
73 Imodel = Idiaest;
74 Iinvmodel = Iinvdiaest;
75 else
76 Imodel = Idia;
77 Iinvmodel = Iinvdia;
78 end
79
80 % Initial Conditions
81 w02 = [0,0,0]; % Initial platform body rates, rad/s
82 q0 = [0,0,0]; % Initial quaternion vector
83 q40 = (1-norm(q0)^2)^.5; % Initial scalar part
84 Wheels0 = biasspeed*2*pi/60; % Initial Wheel Speeds, rad/s
85
86 % PRWP Geometry
87 Beta = 28.3*pi/180; % Wheel Inclination Angle, rad
88 betas = [cos(Beta),sin(Beta)]; % Sine and cosine of beta
89 Rws = [cos(Beta),0,-cos(Beta),0;0,cos(Beta),0,-
cos(Beta);sin(Beta),sin
(Beta),sin(Beta),sin(Beta)]; % Wheel Frame to Body Frame
90 BTorque2WTorque = .5*[1,0,.5;0,1,.5;-1,0,.5;0,-1,.5]; % Torque
Distribution
91
92 %% Reaction Wheels
93 Iw = .00053552; % Reaction Wheel Inertia, kg-m^2
94 % Degraded Reaction Wheel Dynamics

```

```

95 if Case == 3;
96 wn4rw = 1; % Natural Frequency
97 zeta4rw = 1.1; % Damping Ratio
98 A4c = [0 1;-wn4rw^2,-2*zeta4rw*wn4rw];
99 B4c = [0;wn4rw^2];
100 C4c = [1,0];
101 D4c = [0];
102 sysC = ss(A4c,B4c,C4c,D4c);% State Space Representation
103 sysD = c2d(sysC,dt);
104 A4 = sysD.a;
105 B4 = sysD.b;
106 C4 = sysD.c;
107 D4 = sysD.d;
108 else
109 % Nominal Reaction Wheel Dynamics
110 A4 = [0,0;0,0]; % Output = Input
111 B4 = [0;0];
112 C4 = [0,0];
113 D4 = [1];
114 sysC = ss(A4,B4,C4,D4);% State Space Representation
115 sysD = c2d(sysC,dt);
116 A4 = sysD.a;
117 B4 = sysD.b;
118 C4 = sysD.c;
119 D4 = sysD.d;
120 end
121 %% Full State Feedback Control
122 % Full State Feedback Gains
123 ts = 20;
124 damping = .75;
125 pcntsettle = 0.01;
126 wn = -log(pcntsettle)/ts/damping;
127 if Case == 2
128 Kfsfb = 2*wn^2*Idiag;
129 Cfsfb = 2*damping*wn*Idiag;
130 else
131 Kfsfb = 2*wn^2*Idiag;
132 Cfsfb = 2*damping*wn*Idiag;
133 end
134
135 %% Adaptive Control
136 % Adaptive Gain ICs and Adaptive Parameters
137 % NDMRAC Gains
138 if Case == 1 % Known Inertia Tensor
139 S210 = [ -0.0041 -0.0031 -0.0041 -0.0004 -0.0003 -0.0004
-0.0008 -0.0005 -0.0007 0.0008 0.0006 0.0008
140 -0.0028 -0.0021 -0.0028 -0.0001 -0.0001 -0.0001
-0.0003 -0.0002 -0.0003 0.0005 0.0003 0.0005
141 -0.0032 -0.0024 -0.0032 -0.0003 -0.0002 -0.0003
-0.0006 -0.0004 -0.0006 0.0005 0.0004 0.0005
142 0.0016 0.0012 0.0016 -0.0009 -0.0007 -0.0009
-0.0018 -0.0014 -0.0018 0.0004 0.0004 0.0004
143 0.0013 0.0010 0.0013 -0.0005 -0.0003 -0.0005
-0.0009 -0.0007 -0.0009 0.0003 0.0003 0.0004
144 0.0017 0.0013 0.0017 -0.0005 -0.0004 -0.0005
-0.0010 -0.0008 -0.0010 0.0005 0.0004 0.0005];
145 S220 = [0.0003 0.0002 0.0003 0.0115

```

```

146 0.0004 0.0003 0.0004 0.0003
147 -0.0010 -0.0007 -0.0010 0.0120
148 -0.0008 -0.0006 -0.0008 -0.0102
149 0.0010 0.0008 0.0010 -0.0000
150 0.0021 0.0016 0.0021 -0.0116];
151 Ge0NDRAC = [-0.6313 -0.0511 -0.0654 -0.0039 0.3784 -0.0686
152 -0.0511 -0.4979 -0.0504 -0.3664 -0.0075 0.1995
153 -0.0654 -0.0504 -0.4494 0.0866 -0.1886 -0.0020
154 -0.0039 -0.3664 0.0866 -1.2362 0.0055 -0.0156
155 0.3784 -0.0075 -0.1886 0.0055 -1.2599 0.0509
156 -0.0686 0.1995 -0.0020 -0.0156 0.0509 -0.9739];
157 else % Unknown Inertia Tensor/Damaged Reaction Wheel
158 S210 = [ -0.0046 -0.0035 -0.0046 -0.0004 -0.0003 -0.0004
-0.0009 -0.0006 -0.0008 0.0011 0.0008 0.0011
159 -0.0030 -0.0023 -0.0031 -0.0002 -0.0001 -0.0002
-0.0004 -0.0003 -0.0004 0.0007 0.0004 0.0007
160 -0.0038 -0.0028 -0.0038 -0.0004 -0.0003 -0.0004
-0.0008 -0.0006 -0.0008 0.0008 0.0006 0.0008
161 0.0014 0.0010 0.0014 -0.0012 -0.0009 -0.0012
-0.0024 -0.0019 -0.0024 0.0007 0.0006 0.0007
162 0.0011 0.0008 0.0011 -0.0007 -0.0005 -0.0007
-0.0013 -0.0010 -0.0013 0.0004 0.0004 0.0005
163 0.0017 0.0013 0.0017 -0.0008 -0.0006 -0.0008
-0.0015 -0.0012 -0.0015 0.0007 0.0006 0.0007];
164 S220 = [ 0.0021 0.0016 0.0021 0.0105
165 0.0011 0.0008 0.0011 0.0040
166 -0.0001 -0.0000 -0.0001 0.0122
167 -0.0006 -0.0005 -0.0006 -0.0101
168 0.0007 0.0005 0.0007 -0.0038
169 0.0022 0.0017 0.0022 -0.0121];
170 Ge0NDRAC = [-0.6461 -0.0572 -0.0781 -0.0041 0.3846 -0.0667
171 -0.0572 -0.5019 -0.0560 -0.3727 -0.0072 0.1974
172 -0.0781 -0.0560 -0.4620 0.0840 -0.1866 -0.0019
173 -0.0041 -0.3727 0.0840 -1.3173 -0.0233 -0.0792
174 0.3846 -0.0072 -0.1866 -0.0233 -1.2946 0.0238
175 -0.0667 0.1974 -0.0019 -0.0792 0.0238 -1.0426];
176 end
177 H1 = .1*eye(12);
178 H2 = .1*eye(4);
179 H3NDRAC = 1000*eye(6);
180 Kadapt = eye(3);
181 Cadapt = eye(3);
182
183 % AOF Gains
184 if Case == 1 % Known Inertia Tensor
185 Ge0AOF = [-1.2665 -0.2727 -0.3673 0.0207 0.1487 -0.0333
186 -0.2727 -0.5781 -0.2971 -0.2026 0.0127 0.1253
187 -0.3673 -0.2971 -0.6832 0.0467 -0.1347 0.0335
188 0.0207 -0.2026 0.0467 -1.3179 -0.4065 -0.3688
189 0.1487 0.0127 -0.1347 -0.4065 -0.7541 -0.3735
190 -0.0333 0.1253 0.0335 -0.3688 -0.3735 -1.0816];
191 else % Unknown Inertia Tensor/Damaged Reaction Wheel
192 Ge0AOF = [-0.6274 -0.1230 -0.2110 0.0087 0.1151 0.0005
193 -0.1230 -0.3673 -0.0916 -0.1308 0.0148 0.0282
194 -0.2110 -0.0916 -0.4764 -0.0065 -0.0412 0.0162
195 0.0087 -0.1308 -0.0065 -0.5172 -0.1732 -0.2124
196 0.1151 0.0148 -0.0412 -0.1732 -0.3631 -0.1882

```

```

197 0.0005 0.0282 0.0162 -0.2124 -0.1882 -0.4660];
198 end
199 H3AOF = 1000*eye(6);
200 Kadapt = eye(3);
201 Cadapt = eye(3);
202 %% Perform Simulation
203 % Run Simulation
204 sim('MattsTests')
205 toc
206 %% Save Data
207 if Save == 1
208 fname=[pname,trial,date,'_Case',num2str(Case),'_Controller',num2str
(Controller),'_mat'];
209 save(fname);
210 end
211

```

A.2 PRWP Computer Simulation m-file

```

1 % PRWP Simulations Setup
2 % Matt Downs
3 % 7-23-09
4 % This m-file sets up the PRWP Computer Simulations
5
6 %global NDMRACErrorGain NDMRACInputGain NDMRACStateGain
7
8 %clear
9 clc
10
11 trial = 'A_';
12 pname = 'G:\Documents\Thesis Documents\Simulations\Saved Simulation
Data\';
13 %% Which Case Are You Doing and With What Controller?
14 % Which cases do you want to run?
15 Cases = [1,2,3];
16 % Case I: Nominal Case. Good RW's and Known Inertia Tensor
17 % Case II: Good RW's and Estimated Inertia Tensor
18 % Case III: RW 4 is bad, Known Inertia Tensor
19 % Which controllers do you want to use?
20 Controllers = [1,2,3];
21 % Controller I: Full State Feedback
22 % Controller II: NDMRAC
23 % Controller III: AOF
24 % Do you want to save the data? Y = 1, N = 0;
25 Save = 1;
26 % Simulation Time, seconds
27 time = 70;
28 dt = .02;
29
30 %% Perform Desired Cases
31 for i = 1:length(Cases)
32 for j = 1:length(Controllers)
33 Case = Cases(i);
34 Controller = Controllers(j);

```

```

35
36 % System Information
37 % PRWP Mass Properties
38 I = [.768,-.0348,-.0355;-.0348,.699,-.0165;-.0355,-.0165,.589]; %
SysID PRWP
Inertia, kg-m^2
39 Iinv = inv(I); % Inverse of inertia tensor
40 Idiag = [I(1,1),0,0;0,I(2,2),0;0,0,I(3,3)]; % Linearized PRWP
Inertia, kgm^
2
41 Iinvdiag = inv(Idiag); % Inverse of linearized inertia tensor
42 Iest = [.575,.0265,-.00234;.0265,.576,.00615;-.00234,.00615,.580]; %
Estimated PRWP Inertia, kg-m^2
43 Iinvest = inv(Iest); % Inverse of estimated inertia tensor
44 Idiagest = [Iest(1,1),0,0;0,Iest(2,2),0;0,0,Iest(3,3)]; % Linearized
PRWP
Estimated Inertia, kg-m^2
45 Iinvdiagest = inv(Idiagest); % Inverse of estimated inertia
linearized
tensor
46 if Case == 2;
47 Imodel = Idiagest;
48 Iinvmodel = Iinvdiagest;
49 %I = Iest;
50 else
51 Imodel = Idiag;
52 Iinvmodel = Iinvdiag;
53 end
54
55 % Inital Conditions
56 w02 = [0,0,0]; % Initial platform body rates, rad/s
57 q0 = [0,0,0]; % Initial quaternion vector
58 q40 = (1-norm(q0)^2)^.5; % Inital scalar par
59 biasspeed = [3800,2200,2200,3800]; % RPM
60 Wheels0 = biasspeed.*2*pi/60; % Initial Wheel Speeds, rad/s
61
62 % PRWP Geometry
63 Beta = 28.3*pi/180; % Wheel Inclination Angle, rad
64 betas = [cos(Beta),sin(Beta)]; % Sine and cosine of beta
65 Rws = [cos(Beta),0,-cos(Beta),0;0,cos(Beta),0,-
cos(Beta);sin(Beta),sin
(Beta),sin(Beta),sin(Beta)]; % Wheel Frame to Body Frame
66 BTorque2WTorque = .5*[1,0,.5;0,1,.5;-1,0,.5;0,-1,.5]; % Torque
Distribution
67
68 %% Reaction Wheels
69 Iw = .00053552; % Reaction Wheel Inertia, kg-m^2
70 % Nominal Reaction Wheel Dynamics
71 wnrw = 8; % Natural Frequency
72 zetarw = 1.1; % Damping Ratio
73 AWc = [0 1;-wnrw^2,-2*zetarw*wnrw];
74 BWc = [0;wnrw^2];
75 CWc = [1,0];
76 DWc = [0];
77 sysC = ss(AWc,BWc,CWc,DWc); % State Space Representation
78 sysD = c2d(sysC,dt);
79 AW = sysD.a;

```

```

80 BW = sysD.b;
81 CW = sysD.c;
82 DW = sysD.d;
83 % Degraded Reaction Wheel Dynamics
84 if Case == 3;
85 wn4rw = 1; % Natural Frequency
86 zeta4rw = 1.1; % Damping Ratio
87 A4c = [0 1;-wn4rw^2,-2*zeta4rw*wn4rw];
88 B4c = [0;wn4rw^2];
89 C4c = [1,0];
90 D4c = [0];
91 sysC = ss(A4c,B4c,C4c,D4c);% State Space Reperesentation
92 sysD = c2d(sysC,dt);
93 A4 = sysD.a;
94 B4 = sysD.b;
95 C4 = sysD.c;
96 D4 = sysD.d;
97 else
98 A4 = [0,0;0,0]; % Output = Input
99 B4 = [0;0];
100 C4 = [0,0];
101 D4 = [1];
102 sysC = ss(A4,B4,C4,D4);% State Space Reperesentation
103 sysD = c2d(sysC,dt);
104 A4 = sysD.a;
105 B4 = sysD.b;
106 C4 = sysD.c;
107 D4 = sysD.d;
108 end
109 %% Full State Feedback Control
110 % Full State Feedback Gains
111 ts = 20;
112 damping = .75;
113 pcntsettle = 0.01;
114 wn = -log(pcntsettle)/ts/damping;
115 if Case == 2
116 Kfsfb = 2*wn^2*Idiagest; % Full State Feedback Gain K
117 Cfsfb = 2*damping*wn*Idiagest; % Full State Feedback Gain C
118 else
119 Kfsfb = 2*wn^2*Idiag; % Full State Feedback Gain K
120 Cfsfb = 2*damping*wn*Idiag; % Full State Feedback Gain C
121 end
122
123 %% Adaptive Control
124 % Adaptive Gain ICs and Adaptive Parameters
125 % NDMRAC Gains
126 if Case == 1 % Known Inertia Tensor
127 %S210 = 0*ones(6,12);
128 S210 = [ -0.0041 -0.0031 -0.0041 -0.0004 -0.0003 -0.0004
-0.0008 -0.0005 -0.0007 0.0008 0.0006 0.0008
129 -0.0028 -0.0021 -0.0028 -0.0001 -0.0001 -0.0001
-0.0003 -0.0002 -0.0003 0.0005 0.0003 0.0005
130 -0.0032 -0.0024 -0.0032 -0.0003 -0.0002 -0.0003
-0.0006 -0.0004 -0.0006 0.0005 0.0004 0.0005
131 0.0016 0.0012 0.0016 -0.0009 -0.0007 -0.0009
-0.0018 -0.0014 -0.0018 0.0004 0.0004 0.0004
132 0.0013 0.0010 0.0013 -0.0005 -0.0003 -0.0005

```



```

-0.0009 -0.0007 -0.0009 0.0003 0.0003 0.0004
133 0.0017 0.0013 0.0017 -0.0005 -0.0004 -0.0005
-0.0010 -0.0008 -0.0010 0.0005 0.0004 0.0005];
134 %S220 = 0*ones(6,4);
135 S220 = [0.0003 0.0002 0.0003 0.0115
136 0.0004 0.0003 0.0004 0.0003
137 -0.0010 -0.0007 -0.0010 0.0120
138 -0.0008 -0.0006 -0.0008 -0.0102
139 0.0010 0.0008 0.0010 -0.0000
140 0.0021 0.0016 0.0021 -0.0116];
141 %Ge0NDRAC = 0*ones(6,6);
142 Ge0NDRAC = [-0.6313 -0.0511 -0.0654 -0.0039 0.3784 -0.0686
143 -0.0511 -0.4979 -0.0504 -0.3664 -0.0075 0.1995
144 -0.0654 -0.0504 -0.4494 0.0866 -0.1886 -0.0020
145 -0.0039 -0.3664 0.0866 -1.2362 0.0055 -0.0156
146 0.3784 -0.0075 -0.1886 0.0055 -1.2599 0.0509
147 -0.0686 0.1995 -0.0020 -0.0156 0.0509 -0.9739];
148 else % Unknown Inertia Tensor/Damaged Reaction Wheel
149 %S210 = 0*ones(6,12);
150
151 S210 = [ -0.0046 -0.0035 -0.0046 -0.0004 -0.0003 -0.0004
-0.0009 -0.0006 -0.0008 0.0011 0.0008 0.0011
152 -0.0030 -0.0023 -0.0031 -0.0002 -0.0001 -0.0002
-0.0004 -0.0003 -0.0004 0.0007 0.0004 0.0007
153 -0.0038 -0.0028 -0.0038 -0.0004 -0.0003 -0.0004
-0.0008 -0.0006 -0.0008 0.0008 0.0006 0.0008
154 0.0014 0.0010 0.0014 -0.0012 -0.0009 -0.0012
-0.0024 -0.0019 -0.0024 0.0007 0.0006 0.0007
155 0.0011 0.0008 0.0011 -0.0007 -0.0005 -0.0007
-0.0013 -0.0010 -0.0013 0.0004 0.0004 0.0005
156 0.0017 0.0013 0.0017 -0.0008 -0.0006 -0.0008
-0.0015 -0.0012 -0.0015 0.0007 0.0006 0.0007];
157 %S220 = 0*ones(6,4);
158 S220 = [ 0.0021 0.0016 0.0021 0.0105
159 0.0011 0.0008 0.0011 0.0040
160 -0.0001 -0.0000 -0.0001 0.0122
161 -0.0006 -0.0005 -0.0006 -0.0101
162 0.0007 0.0005 0.0007 -0.0038
163 0.0022 0.0017 0.0022 -0.0121];
164 %Ge0NDRAC = 0*ones(6,6);
165 Ge0NDRAC = [-0.6461 -0.0572 -0.0781 -0.0041 0.3846 -0.0667
166 -0.0572 -0.5019 -0.0560 -0.3727 -0.0072 0.1974
167 -0.0781 -0.0560 -0.4620 0.0840 -0.1866 -0.0019
168 -0.0041 -0.3727 0.0840 -1.3173 -0.0233 -0.0792
169 0.3846 -0.0072 -0.1866 -0.0233 -1.2946 0.0238
170 -0.0667 0.1974 -0.0019 -0.0792 0.0238 -1.0426];
171
172 end
173 H1 = .1*eye(12);
174 H2 = .1*eye(4);
175 H3NDRAC = 1000*eye(6);
176 Kadapt = eye(3);
177 Cadapt = eye(3);
178
179 % AOF Gains
180 if Case == 1 % Known Inertia Tensor
181 %Ge0AOF = 0*ones(6,6);

```

```

182 GeAOF = [-1.2665 -0.2727 -0.3673 0.0207 0.1487 -0.0333
183 -0.2727 -0.5781 -0.2971 -0.2026 0.0127 0.1253
184 -0.3673 -0.2971 -0.6832 0.0467 -0.1347 0.0335
185 0.0207 -0.2026 0.0467 -1.3179 -0.4065 -0.3688
186 0.1487 0.0127 -0.1347 -0.4065 -0.7541 -0.3735
187 -0.0333 0.1253 0.0335 -0.3688 -0.3735 -1.0816];
188 else % Unknown Inertia Tensor/Damaged Reaction Wheel
189 %Ge0AOF = 0*ones(6,6);
190 Ge0AOF = [-0.6274 -0.1230 -0.2110 0.0087 0.1151 0.0005
191 -0.1230 -0.3673 -0.0916 -0.1308 0.0148 0.0282
192 -0.2110 -0.0916 -0.4764 -0.0065 -0.0412 0.0162
193 0.0087 -0.1308 -0.0065 -0.5172 -0.1732 -0.2124
194 0.1151 0.0148 -0.0412 -0.1732 -0.3631 -0.1882
195 0.0005 0.0282 0.0162 -0.2124 -0.1882 -0.4660];
196 end
197 H3AOF = 1000*eye(6);
198 Kadapt = eye(3);
199 Cadapt = eye(3);
200 %% Perform Simulation
201 % Run Simulation
202 sim('PRWP_Simulation',time)
203
204 %% Save Data
205 if Save == 1
206 fname=[pname,trial,date,'_Case',num2str(Case),'_Controller',num2str
207 (Controller),'.mat'];
208 save(fname);
209 end
210 end
211

```

B.1 SADS Simulation Simulink Block Diagrams

SADS/ACF PRWP Control/"Square" the Plant

SADS/AOF PRWP Control/Adaptive Error Gain

SADS/AOF PRWP Control/Plant Outputs

SADS/AOF PRWP Control/Reference Model1/Desired Torque to Desired Wheel Speeds/Cross Product.eML_blk_kernel

```
1: function u1Xu2 = Xproduct(u1,u2)
2: % This block supports the Embedded MATLAB subset.
3: % See the help menu for details.
4: % Computes the cross product, u1 X u2, of the 3X1 vectors u1 and u2.
5:
6: u1Xu2= [0,-u1(3),u1(2);u1(3),0,-u1(1);-u1(2),u1(1),0]*[u2(1);u2(2);u2(3)]
```

%<fullsystem>

SADS/AOF PRWP Control/Reference Model1/Desired Torque to Desired Wheel Speeds/Tprime.eML_blk_kernel

```
1: function Tprime = Tprime(T,betas)
2: % This block supports the Embedded MATLAB subset.
3: % See the help menu for details.
4: % Computes the cross product, u1 X u2, of the 3X1 vectors u1 and u2.
5:
6: Tprime = [T(1)/betas(1);T(2)/betas(1);T(3)/betas(2)];
```

%<fullsystem>

SADS/AOF PRWP Control/Reference Model1/FSFB PRWP Control

```

1: function qe = ErrQuat(q,qc)
2: % This block supports the Embedded MATLAB subset.
3: % See the help menu for details.
4:
5: qer = [qc(4),qc(3),-qc(2),-qc(1);-qc(3),qc(4),qc(1),-qc(2);qc(2),-qc(1),qc(4),-qc(3);qc(1),qc(2),qc(3),qc(4)]*[q(1);q(2);q(3);q(4)];
6: qe = qer(1:3);

```

SADS/AOF PRWP Control/Reference Model/PRWP Dynamics

SADS/AOF PRWP Control/Reference Model 1/Quaternions Kinematics

SADS/Meas Wheel Rates

SADSNDMRAC PRWP Control

SADS/NDMRAC PRWP Control/Adaptive Input Gain

SADS/NDMRAC PRWP Control/Adaptive State Gain

Reaction Wheel Dynamics

SADS/NDMRAC PRWP Control/Reference Model/Reaction Wheel Dynamics

SADS/Track Platform

SADS/Wheel Kalman Filters

B.2 PRWP Computer Simulation Simulink Block Diagrams

Note: Many subsystems in the PRWP Computer Simulation are the same as subsystems in the SADS Simulation, so they are omitted here.

PRWP_Simulation/Reaction Wheel Dynamics

Appendix C: DACC Interface Electronics Schematics

The schematic of the power board is shown in Figure C.1. Power is supplied to the board via one of the 18 Volt batteries. Physical switches turn the board on and off. Diodes are in place to protect the voltage regulators from a reverse voltage failure.

The 12 Volt output is created from a positive 12V linear voltage regulator. The 12V output is divided down to 2.5V via a resistive voltage divider. The 2.5V node is then fed through an op-amp voltage follower. This is done because the 2.5V node needs to allow current to flow opposite the direction that the 12V regulator will allow. Alternately, a negative voltage regulator could produce 2.5V in addition to the positive 12V regulator. Finally, an LED is used to indicate that the 12V regulator is operational.

Three positive 5V linear voltage regulators create three independent 5V outputs. After being regulated down to 5V, the 5V sources are passed through single-pole single-throw, normally closed relays. These relay function as a safeguard in the event that the 5V regulators fail and output more than 5V. In the event that 5.5 or more Volts is passed to the inputs of the relay, the relay is switched open and the 5V output is left open. The resistors R_1 , R_2 , and R_3 are used to change the set point of the relays so that they open at 5.5 Volts. The .5 Ohm resistors and 220 μ F capacitors are in place so that the voltage will not rise too high before the relay switch it activated. LED's are in place to indicate that 5V is reaching the outputs.

Figure C.1 Power Board Schematic

The purpose of the frequency to voltage conversion circuit is to transform the frequency of the square wave generated by the motor encoders into a voltage proportional to the frequency. The circuit is similar to the instructions provided by the manufacturer of the LM2907N integrated circuit. A schematic of the circuit board is shown in Figure C.2.

Figure C.2 Frequency to Voltage Conversion Circuit

The PWM generation circuit is designed to create four independent PWM signals whose duty cycles are controlled by input voltages. A schematic of this circuit is shown in Figure C.3. Via a feedback system, the system of two op-amps creates a triangle waveform. This triangle waveform is then compared to the input voltages via op-comparators. If the input voltage is higher than the triangle waveform, the output of comparator has very high impedance to ground. A pull-up resistor is used to pull the voltage up to 5V. If the input voltage is lower than the triangle waveform, the output of the comparator is connected to ground. In this way, a PWM signal with variable duty cycle is created. A pull-down resistor on the input voltages is used to ensure that the PWM has 0% duty cycle if the inputs are disconnected.

Figure C.3 PWM Generation Circuit