Utah State University
DigitalCommons@USU

All Graduate Plan B and other Reports | Graduate Studies

12-2014

Design and Testing of a Nanosatellite Simulator Reaction Wheel Attitude Control System

Fredric William Long
Utah State University

Follow this and additional works at: https://digitalcommons.usu.edu/gradreports

Part of the Mechanical Engineering Commons

Recommended Citation

Long, Fredric William, "Design and Testing of a Nanosatellite Simulator Reaction Wheel Attitude Control System" (2014). *All Graduate Plan B and other Reports*. 448.
https://digitalcommons.usu.edu/gradreports/448

This Thesis is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Plan B and other Reports by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.

DESIGN AND TESTING OF A NANOSATELLITE SIMULATOR REACTION WHEEL ATTITUDE CONTROL SYSTEM

by

Fredric Long

A report submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Mechanical Engineering

Approved:

Dr. Rees Fullmer Major Professor	Dr. David Geller Committee Member

Dr. Aaron Katz
Committee Member

UTAH STATE UNIVERSITY
Logan, Utah

2014

Copyright © Fredric Long 2014

All Rights Reserved

Abstract

Design and Testing of a Nanosatellite Simulator Reaction Wheel Attitude Control System

by

Fredric Long, Master of Science

Utah State University, 2014

Major Professor: Dr. Rees Fullmer
Department: Mechanical and Aerospace Engineering

Attitude control of a satellite is required for the pointing of communications antennas and other instruments. For this reason, a control algorithm capable of precision pointing is important. Because the process of sending a satellite into space is time consuming and costly, ground-based methods of testing are paramount. A simulator is a low-cost, ground-based system made to mimic the conditions of a weightless satellite in space. The simulator used in this project controls attitude through applying torque to reaction wheels. The objective of this project is to derive and test a control algorithm for the attitude control of a satellite simulator.

(73 pages)

Acknowledgments

First, I would like to thank Jan Sommer, Landon Terry and Marina Samuels for their previous work on the simulator. Without their work, my project would not be possible. I would also like to thank my major professor Dr. Rees Fullmer for the opportunity to work on this project and for providing essential feedback throughout the project.

I am also grateful to my family for their momentous support. A special thanks goes out to my wonderful wife Rose for her patience and assistance throughout my education.

Fredric Long

Contents

Page

Abstract . . . . . **iii**

Acknowledgments . . . . . **iv**

List of Tables . . . . . **vii**

List of Figures . . . . . **viii**

Notation . . . . . **x**

1 Introduction **1**
1.1 Overview 1
1.2 Motivation 2
1.3 Objectives 2

2 Literature Review **4**

3 System Description **6**
3.1 Air Bearing 7
3.2 Attitude Sensing 7
3.3 Processor 9
3.4 Motors and Reaction Wheels 10
3.5 Power Supply 11
3.6 Platform 11

4 Simulator Dynamic Model **12**
4.1 System Dynamics 12
4.2 External Torques 15
4.3 Motor Dynamics 15

5 Control Strategy **16**
5.1 Determining Body Axes Torque from Wheel Torques 16
5.2 Controller Design 18
5.3 Quaternion Error 21
5.4 Trajectory Generation 23
5.5 External Motor Controller 24
5.6 Controller Implementation 25
5.7 Linear System Approximation 26
5.8 Nonlinear System Approximation 28

6 Testing **30**
6.1 System Properties 30
6.2 IMU Performance 30
6.3 Motor Response 33
6.4 Stationkeeping 36
6.5 Control of Motion in One Degree of Freedom 41
6.5.1 One Degree of Freedom Test without Trajectory Generation 41
6.5.2 One Degree of Freedom Test with Trajectory Generation 44
6.6 Three Degree of Freedom Response with Trajectory 46

7 Conclusions and Future Work **54**

References **56**

Appendices **57**
A IMU Data Sheet 58
B Motor Data Sheet 61

List of Tables

Table Page

6.1 Table of testing properties 30

6.2 Table of mean and standard deviation for drift test 32

6.3 Table of mean and standard deviation for stationkeeping test with $\omega_n = 1rad/s$ 38

6.4 Table of mean and standard deviation for stationkeeping test with $\omega_n = 2rad/s$ 40

List of Figures

Figure Page

1.1 Top view of the simulator 1

3.1 Top view of the simulator 6

3.2 Air bearing cup 7

3.3 IMU location 8

3.4 Front view of the microcomputer (BB) 9

3.5 Closeup photo of a reaction wheel, motor, and servo controller 10

3.6 View of the underside of the platform while sitting on air bearing 11

4.1 Sketch of the top of the simulator identifying wheel numbering, body frame, and torque directions. 13

5.1 Example of a trapezoidal trajectory 23

5.2 Closed loop system structure 26

5.3 Continuous linear approximation of controlled system dynamics 28

5.4 Continuous nonlinear approximation of controlled system dynamics 28

6.1 Euler angles of a fixed simulator test 31

6.2 Angular velocity of a fixed simulator test 32

6.3 Unfiltered motor response with step input 34

6.4 Angular velocity of the simulator using $\hat{\tau} = 0.1$ 35

6.5 Velocity of wheel 0 using $\hat{\tau} = 0.1$ 35

6.6 Euler angles of the simulator for stationkeeping test with $\omega_n = 1rad/s$ . . . 37

6.7 Velocity of wheel 0 for stationkeeping test with $\omega_n = 1rad/s$ 38

6.8 Euler angles of the simulator for stationkeeping test with $\omega_n = 2rad/s$ . . . 39

6.9 Velocity of wheel 2 for stationkeeping test with $\omega_n = 2rad/s$ 40

6.10 Euler angles of the simulator with $\omega_n = 2rad/s$ for the no trajectory test . 42

6.11 References and measured position in yaw with $\omega_n = 2rad/s$ for the no trajectory test 43

6.12 Velocity of wheel 0 with $\omega_n = 2rad/s$ for the no trajectory test 44

6.13 Euler angles showing control in yaw using trajectory generation 45

6.14 Reference, trajectory, and measured angular position in yaw using trajectory generation 46

6.15 References, trajectory, and measured position in roll with $\omega_n = 1rad/s$ for three degree of freedom test 47

6.16 References, trajectory, and measured position in pitch with $\omega_n = 1rad/s$ for three degree of freedom test 48

6.17 References, trajectory, and measured position in yaw with $\omega_n = 1rad/s$ for three degree of freedom test 49

6.18 Velocity of wheel 0 with $\omega_n = 1rad/s$ for three degree of freedom test . . . 50

6.19 References, trajectory, and measured position in roll with $\omega_n = 2rad/s$ for three degree of freedom test 51

6.20 References, trajectory, and measured position in pitch with $\omega_n = 2rad/s$ for three degree of freedom test 52

6.21 References, trajectory, and measured position in yaw with $\omega_n = 2rad/s$ for three degree of freedom test 53

Notation

h	Angular momentum vector
ω	Angular velocity vector of the simulator
Ω	Angular velocity vector of wheels in each wheel's reference frame
G	Transformation matrix from wheel reference frames into body reference frame
H	Transformation matrix G with appended constraint
u	Wheel angular acceleration input vector
J	Moment of inertia
K	Scalar controller gain
Φ	Angular position vector of the simulator in the body frame
τ	Motor time constant
I	Identity matrix
c	Cost function
s	Laplace variable

Subscript

T	Property of the simulator body including the momentum wheels
s	Property of the simulator body
w	Property of the momentum wheels
d	Desired frame or value
e	Error between desired and current

Superscript

I	Property viewed in the inertial frame, centered at the center of mass, fixed to the air bearing cup
B	Property viewed in the body frame, centered at the center of mass, fixed to simulator
w	Property viewed in a wheel frame, centered at the center of mass, fixed to simulator, with an axis parallel to the wheel's axis of rotation
$\hat{}$	Estimated value

Chapter 1

Introduction

1.1 Overview

Attitude control of satellites is needed for pointing research payloads, navigational instruments, communications antennae, and solar panels. This project designs and tests the performance of an attitude control algorithm on a tabletop simulator. The simulator provides an inexpensive platform for ground-based testing while imitating the torque-free conditions of a satellite in space. Figure (1.1) shows the components of the tabletop simulator. The tabletop simulator (sim table) includes all of the components typically found in

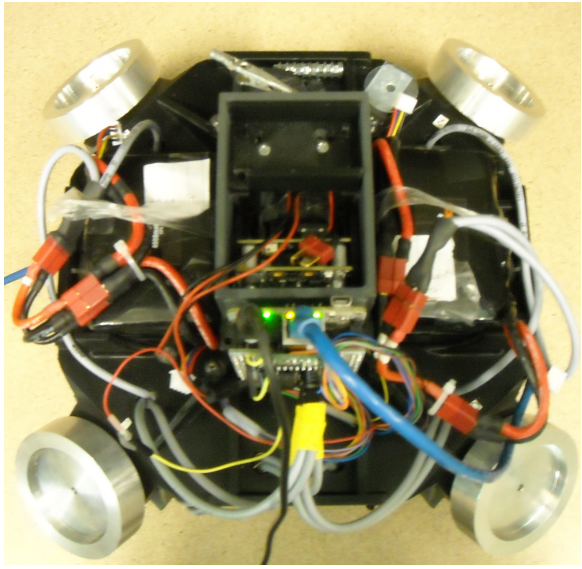


Fig. 1.1: Top view of the simulator

an attitude control system of a spacecraft. This includes reaction wheels, platform, motors with servo controllers, an Inertial Measurement Unit, an on-board microcomputer, batteries, and an air bearing. Control of the simulator will be actuated through the reaction wheels. Angular momentum is transferred between the wheels and the platform resulting

in the desired attitude of the platform. Because the hardware of the simulator was essentially in place at the beginning of the project, this project focuses on the development, testing, and performance of the attitude determination and control system (ADCS) control algorithms.

1.2 Motivation

Because of the relatively small moment of inertia and available power of nanosatellites, attitude control is particularly difficult [1]. Currently, satellite design is trending toward nanosatellites primarily because of their lower cost and ease of launch. The simulator models a nanosatellite since it has a mass of about 7 kg..

While many attitude actuators are available, reaction wheels have many advantages. Reaction wheels offer precision torque generation with relatively low weight and high sensitivity when compared to thrusters. Control moment gyros are used for larger, more agile satellites but have yet to be developed for nanosatellites. Torque coils and rods use the earth's magnetic field as an actuator. However, the magnitude of the control torque is too small for most applications.

Ground-based testing is an important step in small satellite development. Swartwout comments about CubeSat failures between 2000-2012, "Failure rates approaching 50% among university-led missions is distressing, especially since many of the observed failures could be identified and corrected before launch." Of Cubesat failures, 37% are attributed to failures in the power, software, ADCS, Central Processing Unit, and mechanical systems [2]. All of these systems will be tested for this project. It is clear from Swartwout's data that ground-based system-level testing is very important to the success of a mission.

1.3 Objectives

The objective of this project is not to revisit control methods but instead to design a controller for this particular system and demonstrate its performance. This simulator had previously been developed by other students at Utah State University [3], [4]. At the start of this project, the hardware of the simulator was largely in place but the software relating

to the ADCS required changes. Because of this, the identification of the dynamics of the system, development of the controller, and testing of the control system will be the focus of this paper.

- Developing Mathematic Model of the System Dynamics

  A system model will be developed for simulator control with reaction wheels. This is done in order to design a controller and also to compare system performance to the model. Model development involves finding the angular acceleration of the simulator in terms of the wheel velocities. This model is deterministic; it does not take into account any stochastic elements such as sensor noise.

- Design of the Controller

  Based on the system model, a controller is designed such that the simulator follows the desired trajectory. The motives for controller design as well as its implementation will be discussed. The controller uses feedback linearization as well as a Proportional Integral Derivative controller. Trajectory generation is also used in order to improve the system response.

- Testing Hardware Performance

  The performance of the motor and the IMU will be tested separately. A step input will be commanded of the motor. The motor response will be identified and compared to the motor model. The IMU will be tested for position and velocity determination accuracy by collecting data from the simulator when it is immobile. Reasons for inaccuracies such as drift will be discussed.

- Testing of System Dynamic Performance

  The sim table response will be tested. A stationkeeping test will be presented which evaluates the simulator's steady state performance. To test the performance of the closed loop system in one dimension, the simulator is given a command in yaw while the desired position in roll and pitch are constant. A three degree of freedom test is also done in which the simulator is commanded to change roll, pitch, and yaw position.

Chapter 2

Literature Review

The problem of using reaction wheels for satellite attitude control has been addressed many times in many well known spacecraft control textbooks such as Sidi [5] and Wertz [6].

Crowell [7] describes many small satellite testing setups as well as developing and testing against a system model for rotational air bearing simulators. The system dynamics are linearized about equilibrium points. Crowell used a Extended Kalman filter using the steady state gains for attitude determination. A PD LQR controller was used to produce the control input. Feedforward control was also implemented based on the simulator dynamics. For his sim table tests, a rotational air bearing and a simulator using three reaction wheels were used. An external magnetic field was generated such that the magnetometer and torque coil could be used more effectively. Two Arduinos were used as processors. One processor was used for the ACDS and the other processed and output the motor encoder values. His computer simulation response was compared to the actual sim table response. Using momentum dumping, the responses match fairly closely. An estimate of the external torque disturbance was used to adjust the location of the center of mass. He also describes and tests for many sources of external disturbances.

Prado [8] approached the problem of statically and dynamically balancing a tabletop simulator. Two methods of of automatic balancing were compared. The system used masses which could be moved by a motor. He also quantified external torques present during testing.

Janson [9] describes the experience of testing and launching pico-satellites. Attitude of the satellites was actuated using both reaction wheels and magnetic torque coils. He notes that sufficient ground-testing time for attitude control systems are a much more efficient use of time than testing in-flight. He also notes that simplifying the moment of inertia matrix to a diagonal matrix may cause attitude control problems. Momentum dumping every two

minutes was implemented in order to prevent the buildup of angular momentum.

Kang's [10] paper describes optimal attitude control algorithms as well as coordination strategies for multiple satellites. He designed a nonlinear H_∞ optimal controller as well as a controller combining the methods of LQR and sliding mode control.

Sommer [3] developed much of the software for the simulator table at Utah State University (USU). Sommer's paper describes much of the code relating to the data handling system and a star camera. Electrical components of the simulator are documented in the thesis. The design decisions such as electrical architecture, software structure, and processor choice are discussed. The potential use of an external motor controller was suggested.

Samuels [4] was another student who previously worked on the development of the simulator table at USU. Samuels set up much of the controller code and performed component testing. Reasons for the simulator design, software development, and a detailed hardware description is covered in Samuels' paper. A PD controller with trajectory generation was developed. Control in yaw was demonstrated. However, control in roll and pitch were not achieved. Testing of the motors and their responses, as well as filtering and magnetometer accuracy was demonstrated. Samuels also lists many similar historical university simulator tests.

Chapter 3

System Description

The simulator at Utah State University consists of an air bearing, an Inertial Measurement Unit (IMU), on-board microcomputer, four motors and reaction wheels, batteries, and a platform. The IMU processes and sends the sensed angular position and rates to the microcomputer. Based on the current angular position and rates, the controller determines commands to the reaction wheels such that the simulator approaches the desired angular position and rates. Figure (3.1) shows the top view of the simulator with labeled components.

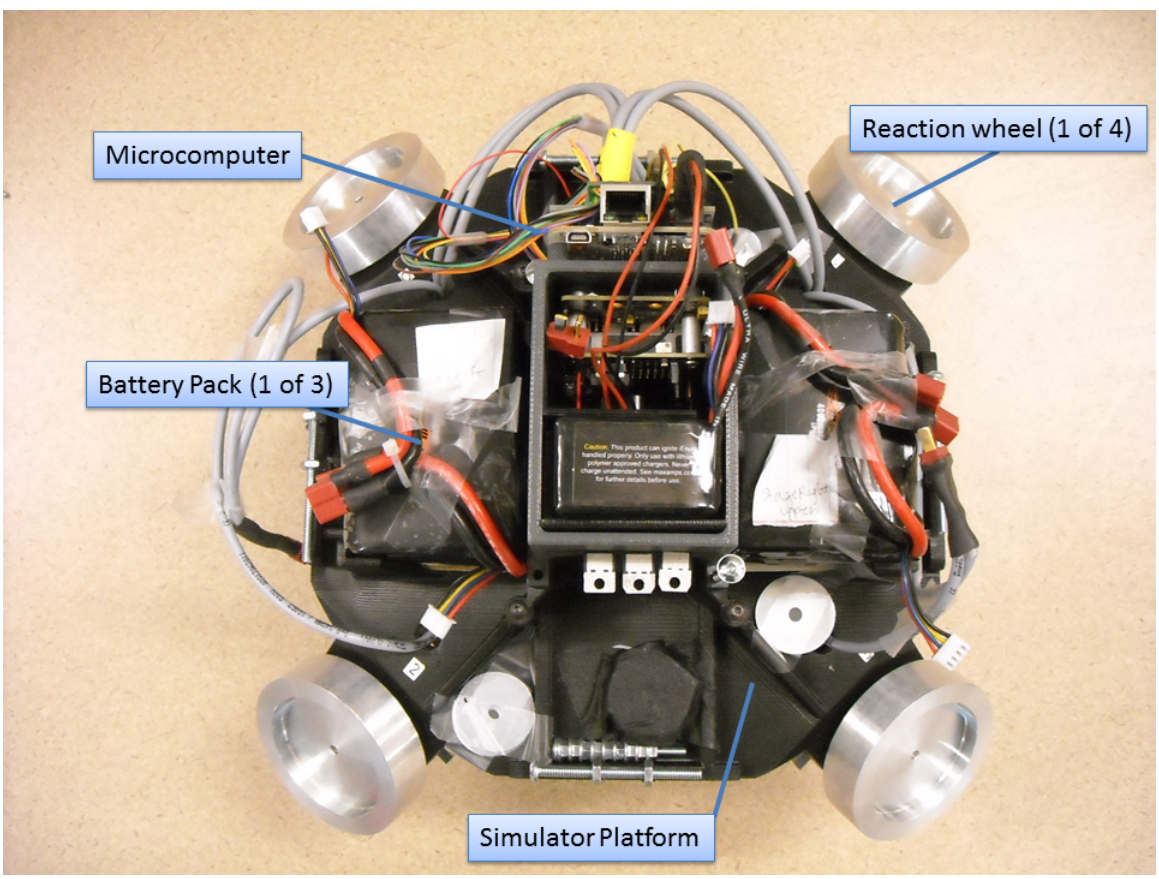


Fig. 3.1: Top view of the simulator

This simulator was designed with the goal of using commercial off-the-shelf (COTS) parts.

3.1 Air Bearing

Many testing methods exist which simulate weightlessness; however, these methods are often not as effective as using an air bearing because of additional forces that arise from friction and countering gravity [11]. The purpose of the air bearing is to allow rotation about all three axes without producing significant external torques. The air bearing consists of a hemisphere and a cup. The hemisphere is a brass half-sphere attached to the bottom of the simulator table which fits into the cup. An air compressor forces air through the cup. The cup shown in fig.(3.2) has six small holes where air is pushed out. This keeps the entire simulator table suspended on a cushion of air and allows the simulator to rotate 360° in yaw and ± 45° in roll and pitch. A large advantage of using an air bearing is the low friction between the hemisphere and the cup.

Fig. 3.2: Air bearing cup

3.2 Attitude Sensing

The Inertial Measurement Unit (IMU) measures, processes, and sends data regarding the orientation of the simulator. The LORD Microstrain 3DM-GX3-25 used consists of a

triaxial accelerometer, triaxial gyro, triaxial magnetometer, and temperature sensors. A product data sheet of the IMU can be found in Appendix A [14]. The mounting location of the IMU can be seen in fig.(3.3). The attitude of the IMU is measured relative to gravity and the earth's magnetic field. The angular rates of the simulator come from the gyro measurement in the body coordinates. A combination of the gyro, gravity, and magnetometer measurements are used internally by the IMU to improve angular position accuracy. This is called north-up compensation. It also implements a moving average finite input response filter to improve data estimates.

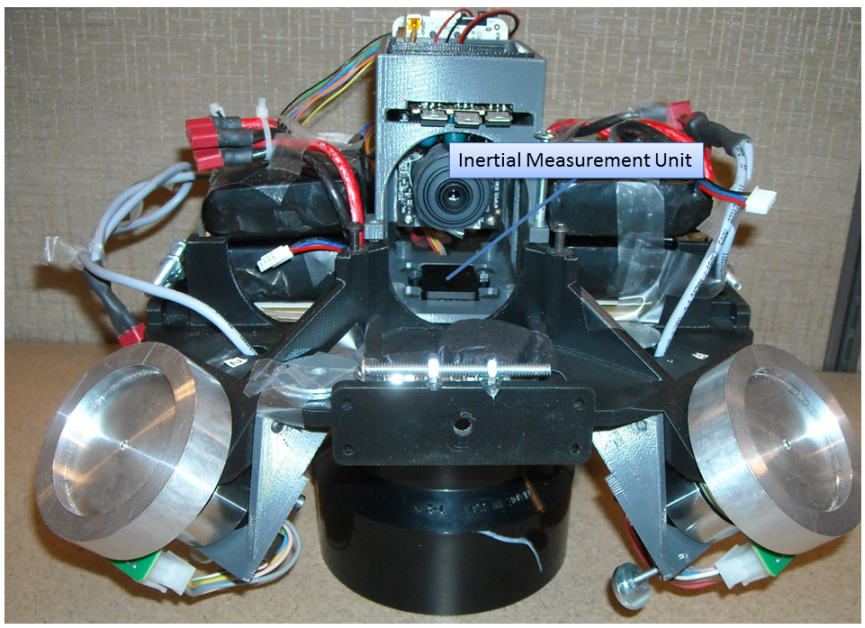


Fig. 3.3: IMU location

Five temperature sensors compensate for the thermal effects on the sensors in the IMU. Because the IMU uses temperature compensation and combines the information from sensors for a better estimate, the angular positions and rates output by the sensors were assumed to be reliable. However, in addition to the IMU filter, the gyro rates were again filtered externally by the controller. The magnetometer had undergone soft and hard iron calibration before the start of this project. Soft and hard iron calibration is needed to compensate for ferro-magnetic components that affect the magnetometer.

3.3 Processor

The purpose of the processor is to import data from the IMU, import data from the motors, compute the output wheel speeds, export the desired wheel speeds, and save the resulting data. It does all of this in real time with a sampling period of 20 milliseconds. A Beagle Bone (BB) microcomputer is included on board the simulator as shown in fig. (3.4). The BB runs a Linux operating system. Code was compiled and run on the BB itself but was commanded from another computer connected by an ethernet cable. The code was set to run after a fixed amount of time, then the commanding computer was detached. In this way, the simulator was tested without any attached cables. The IMU has many output options, but Euler angles and body rates outputs were used.

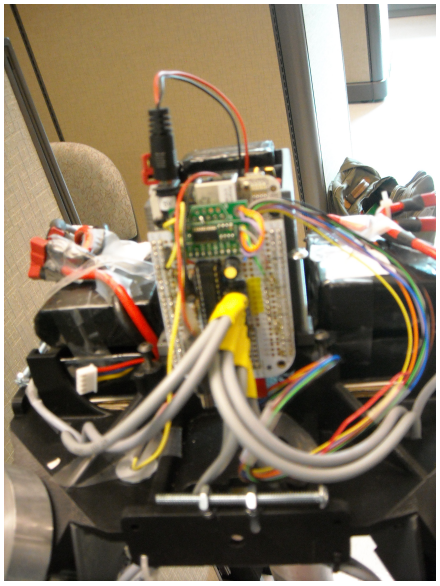


Fig. 3.4: Front view of the microcomputer (BB)

The IMU was mounted upside down because of physical constraints. The controller code rotates the GX-3 coordinate system into the body fixed NED coordinate system. The controller code also performs trajectory generation. The controller filtered both angular rates from IMU gyros and motor velocities using the same first order filter whose cutoff frequency is found in table (6.1).

3.4 Motors and Reaction Wheels

Figure (3.5) shows a closeup view of the reaction wheel, motor, and servo controller. The motors used are DC brushless Maxon EC 45 flat with Hall sensor Model #397172. A datasheet for this motor can be found in Appendix B [15]. This motor was used alongside the ESCON 36/3 EC servo controller. The servo controller is set to the closed loop operating mode which means that the servo controller uses current and velocity sensor data to improve motor response. The wheel velocity is sent to the servo controller by the microcomputer though Pulse Width Modulation (PWM) at a frequency of $53.6kHz$. Using the servo controller's analog output, the wheel speed resolution is $7.7\frac{rad}{s}$.

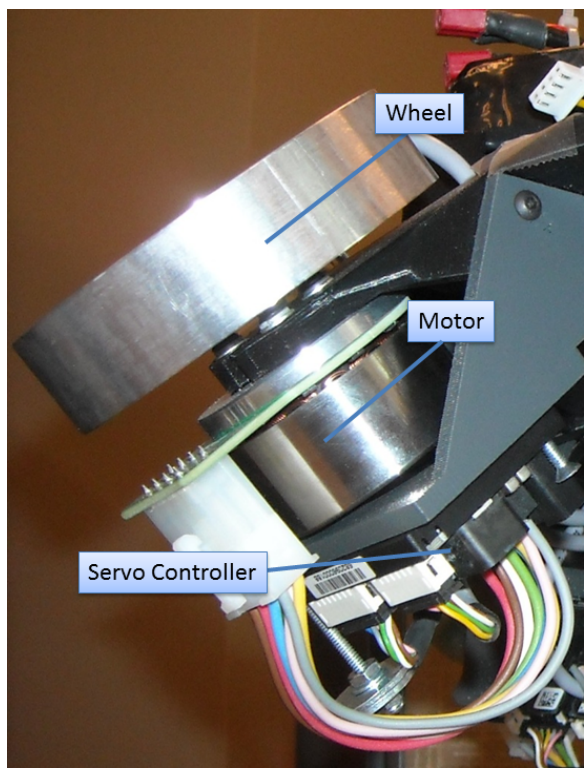


Fig. 3.5: Closeup photo of a reaction wheel, motor, and servo controller

The four reaction wheels are arranged in a pyramidal configuration as shown in fig.(1.1). Based on the density of aluminum and the wheel geometry, each wheel has a moment of inertia of about $1.53e^{-4}kg - m^2$ including the motor shaft inertia. An accurate estimate of this moment of inertia is important for precision control.

3.5 Power Supply

The motors and processor are powered using lithium polymer batteries. Batteries are used instead of a power cord because even the small weight of a power cord may cause the simulator to become unbalanced.

3.6 Platform

The platform rigidly attaches simulator components. The platform is bolted to the brass hemisphere as shown in fig.(3.6). The hemisphere is the only portion of the simulator which comes in contact with the air bearing cup under normal operating conditions. The platform was designed such that the center of mass would be close to the center of rotation. The simulator platform was created by a 3D printer and is made of Acrylonitrile Butadiene Styrene (ABS) plastic.

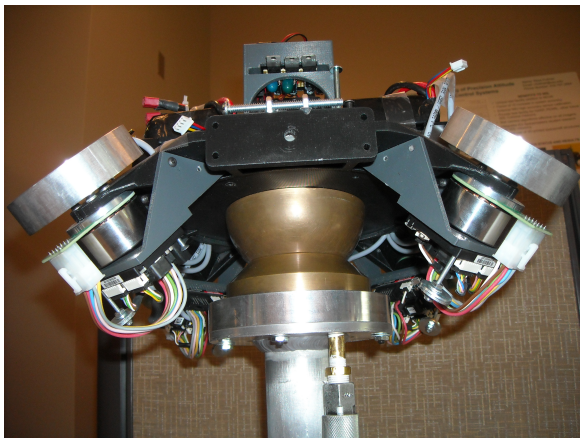


Fig. 3.6: View of the underside of the platform while sitting on air bearing

Only a few hardware modifications had to be made from the previous hardware design. Washers were attached to the platform in order to achieve better balance. The batteries had to be moved vertically in order to bring the center of mass closer to the center of rotation. Screws were added in the x-y plane so that the simulator would be easy to balance. Making the sim table easy to balance was very important since even moving the battery cables could cause the table to lean to one side on the air bearing. For a comprehensive description of this system's components including software see Samuels [4].

Chapter 4

Simulator Dynamic Model

The mathematic system model gives insight into the dynamics of the system and indicates how to select a control law. The goal of this chapter is to find an expression for the torque on the simulator in terms of the angular momenta of the simulator and reaction wheels in the body frame. With this relation, many of sensor outputs may be directly used in a controller since the sensors and actuators are fixed to the body frame. This system will be modeled as a deterministic system.

4.1 System Dynamics

This derivation of the system dynamics follows the derivation presented by Sidi [5]. As stated in eqn.(4.1), the angular acceleration in the inertial frame is equal to the external torques. The total angular acceleration of the system can be split into the contribution of the simulator and that of the reaction wheels.

$$T_e = \dot{h_T^I} = \dot{h_s^I} + \dot{h_w^I} \tag{4.1}$$

Where the definition of angular momentum, h, is

$$h = J\omega \tag{4.2}$$

and T_e is the external torque.

Using Euler's equation for rigid bodies, eqn.(4.1) becomes

$$T_e = \dot{h_s^B} + \omega \times h_s^B + \dot{h_w^B} + \omega \times h_w^B \tag{4.3}$$

It is convenient to rotate the angular momentum of the wheels from the simulator reference

frame into each wheel's reference frame using matrix G. This is so that the angular momentum will be aligned with the wheels' axis of rotation. Notice that the wheel frames and the body frame are all fixed to the simulator. Only a static rotation is required to transform from the body frame to the wheel frame.

The transformation matrix G can be determined from the orientation of the wheels relative to the body frame. Figure (4.1) shows the pyramidal orientation of the wheels and assumed torque directions. The wheels are numbered in this way simply because that is

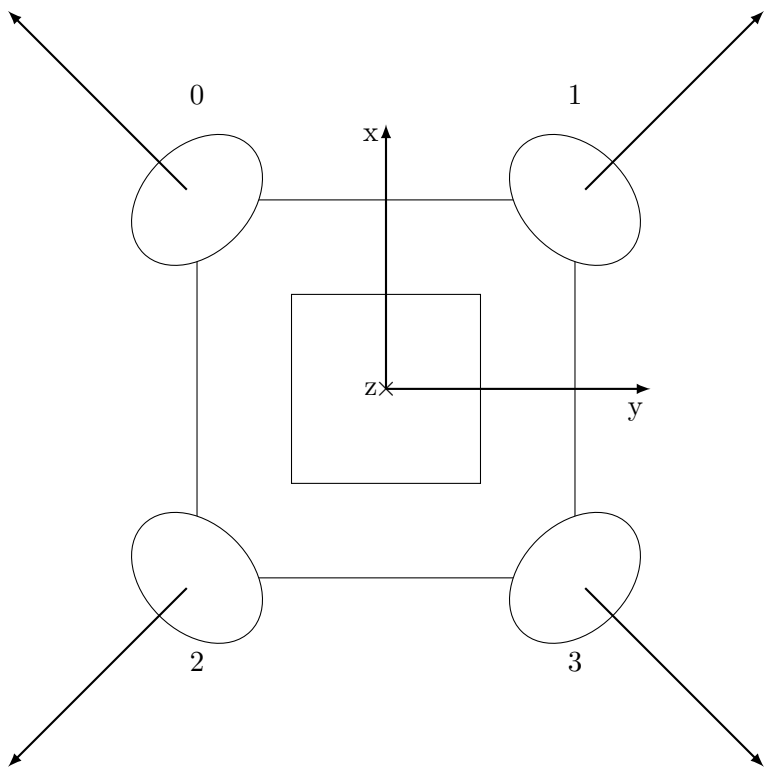


Fig. 4.1: Sketch of the top of the simulator identifying wheel numbering, body frame, and torque directions.

the way the software for the system had been set up. Given that the wheels all point 45° in the negative z direction, the transformation matrix for this setup is

$$G = \begin{bmatrix} \frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} \\ -\frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \\ -\frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} \end{bmatrix} \tag{4.4}$$

The first column of G describes the positive velocity of wheel 0 as having a positive x axis, and negative y and z axis components. It is important to note that the L_2 norm of each column of G must equal unity. For example, when using G to transform the wheel torques into torques on the simulator, the matrix representation is $T_s = GT_w$. Since the Pythagorean Theorem ($T_0^2 = T_{0x}^2 + T_{0y}^2 + T_{0z}^2$) must hold for each wheel, the L_2 norm of each column of G must equal one.

If each wheel has the same moment of inertia about its axis of rotation,

$$h_w^B = J_w^B \omega + G J_w^w \Omega \tag{4.5}$$

Where

$$\Omega = \begin{bmatrix} \Omega_0 \\ \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} \tag{4.6}$$

Where J_w^w is the scalar moment of inertia of one wheel about its axis of rotation and Ω is the velocity of each wheel in its wheel frame. Equation (4.6) could be altered to accommodate an arbitrary number of wheels.

Using the definition $J_s^B + J_w^B = J_T^B$, combining eqn.(4.3), and eqn.(4.5) results in eqn. (4.7).

$$T_e = \dot{h_T^B} + \omega \times h_T^B + G J_w^w \dot{\Omega} + \omega \times G J_w^w \Omega \tag{4.7}$$

Equation (4.7) can be rewritten as

$$\dot{h_T^B} = T_e - \omega \times h_T^B - G J_w^w u - \omega \times G J_w^w \Omega \tag{4.8}$$

Where

$$u = \dot{\Omega} \tag{4.9}$$

Equation (4.8) allows the sensor inputs from the body frame to be used directly to find

the behavior of the simulator in the inertial frame. The transformation matrix G cannot simply be inverted to determine the desired input wheel torque. With this configuration of reaction wheels, there is freedom in how to provide the desired torque on the simulator.

4.2 External Torques

In space, external torques on a satellite are due to the earth's magnetic field, aerodynamics, solar radiation, and expulsion of gas or particles. For a quantitative description of external torques at various altitudes, see Prado [8]. The primary external torque on this simulator is due to the center of gravity not being aligned with the center of rotation. Because of this, great care is taken to balance the simulator before each test.

4.3 Motor Dynamics

The dynamics of the servo controller and motor can be simplified to eqn.(4.10).

$$\dot{\Omega} = \frac{1}{\tau}(\Omega_d - \Omega) \tag{4.10}$$

From this equation it is clear that wheel velocity will converge to the commanded velocity but has a first order dynamic.

Chapter 5

Control Strategy

This control strategy focuses strictly on attitude control and does not approach translational motion control. This is because air bearing prevents the simulator from translating.

5.1 Determining Body Axes Torque from Wheel Torques

Since the four momentum wheels are not collinear and not in the same plane, there is freedom in how to control about the three body axes. One additional equation will be chosen in order to determine the control law. The equation chosen is one that minimizes the L_2 norm of the wheel torques in order to decrease power consumption. This equation is derived below using the Lagrangian method of optimization. The cost function is defined as

$$c = T_0^2 + T_1^2 + T_2^2 + T_3^2 \tag{5.1}$$

Where the subscripts refer to the wheel the torque is generated from. The constraint equations describe that the wheels and their orientations must result in the specified torque on the simulator. Torque may be transformed from the wheel axes into the body axis through the relation

$$\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = G \begin{bmatrix} T_0 \\ T_1 \\ T_2 \\ T_3 \end{bmatrix} \tag{5.2}$$

From eqn.(5.2) and the definition of G, the constraint equations are

$$\frac{\sqrt{3}}{3}(T_0 + T_1 - T_2 - T_3) - T_x = 0 \tag{5.3}$$

$$\frac{\sqrt{3}}{3}(-T_0 + T_1 - T_2 + T_3) - T_y = 0 \tag{5.4}$$

$$\frac{\sqrt{3}}{3}(-T_0 - T_1 - T_2 - T_3) - T_z = 0 \tag{5.5}$$

The Hamiltonian is then

$$H = T_0^2 + T_1^2 + T_2^2 + T_3^2 + \lambda_1(\frac{\sqrt{3}}{3}(T_0 + T_1 - T_2 - T_3) - T_x) +$$
$$\lambda_2(\frac{\sqrt{3}}{3}(-T_0 + T_1 - T_2 + T_3) - T_y) + \lambda_3(\frac{\sqrt{3}}{3}(-T_0 - T_1 - T_2 - T_3) - T_z) \tag{5.6}$$

Where λ is the Lagrange multiplier. The necessary conditions for the minimum are then

$$\frac{\partial H}{\partial T_0} = 2T_0 + \frac{\sqrt{3}}{3}(\lambda_1 - \lambda_2 - \lambda_3) = 0 \tag{5.7}$$

$$\frac{\partial H}{\partial T_1} = 2T_1 + \frac{\sqrt{3}}{3}(\lambda_1 + \lambda_2 - \lambda_3) = 0 \tag{5.8}$$

$$\frac{\partial H}{\partial T_2} = 2T_2 + \frac{\sqrt{3}}{3}(-\lambda_1 - \lambda_2 - \lambda_3) = 0 \tag{5.9}$$

$$\frac{\partial H}{\partial T_3} = 2T_3 + \frac{\sqrt{3}}{3}(-\lambda_1 + \lambda_2 - \lambda_3) = 0 \tag{5.10}$$

Solving this system of equations by eliminating λ_1, λ_2, λ_3 in eqn.(5.7-5.10), the minimum torque equation is

$$T_0 - T_1 - T_2 + T_3 = 0 \tag{5.11}$$

This result is only a static minimum; it does not take into account the dynamics of the system to find the minimum. For a description of reaction wheel regenerative power optimal control, see Blenden [1].

This additional constraint can be used alongside the physical constraint by adjoining the matrix G with eqn.(5.11) to create the matrix H .

$$H = \begin{bmatrix} \frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} \\ -\frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \\ -\frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{5.12}$$

Using this result, the desired wheel angular accelerations u_d can be derived from the desired body axis torques T_d. That is to say

$$\dot{\Omega}_d = u_d = \frac{H^{-1}}{J_w^w} \begin{bmatrix} T_d \\ 0 \end{bmatrix} = \frac{H^{-1}}{J_w^w} \begin{bmatrix} GJ_w^w u_d \\ 0 \end{bmatrix} \tag{5.13}$$

Note that this result is equivalent to method using the right pseudoinverse of G presented by Wertz [6].

5.2 Controller Design

The design of this nonlinear controller will be based on the developed system model. The controller presented implements feedback linearization to make nonlinear system dynamics negligible. Feedback linearization is the method of choosing an input that cancels nonlinear system dynamics. The resulting linear system is controlled using a Proportional Integral Derivative (PID) controller. With the assumption that the external torque is considered negligible,

$$T_e = 0 \tag{5.14}$$

Using this and eqn.(4.8) leaves the equation

$$\dot{h}_T^B = -T_d - \omega \times GJ_w^w \Omega - \omega \times h_T^B \tag{5.15}$$

Where

$$T_d = GJ_w^w u_d \tag{5.16}$$

Therefore, the feedback linearization portion of the control law is chosen as

$$T_{FL} = -\hat{\omega} \times G\hat{J}_w^w \hat{\Omega} - \hat{\omega} \times \hat{h_T^B} \tag{5.17}$$

Where ˆ denotes a measured value. The resulting system dynamics are

$$\dot{h_T^B} = -T_{PID} + (\hat{\omega} \times G\hat{J}_w^w \hat{\Omega} - \omega \times GJ_w^w \Omega) + (\hat{\omega} \times \hat{h_T^B} - \omega \times h_T^B) \tag{5.18}$$

Assuming the errors in the observed values are negligible,

$$\dot{h}_T^B = -T_{PID} \tag{5.19}$$

The chosen PID portion of the control law is then

$$T_{PID} = -\hat{J}_T(K_i \int \Phi_e dt + K_p(\Phi_e) + K_d(\dot{\Phi}_e) + \ddot{\Phi}_d) \tag{5.20}$$

Where Φ is the angular position of the simulator derived from the quaternion position and the subscript e denotes the error between the body frame and the desired frame. The gains K_d, K_p, and K_i are scalar values. Instead of $J_s K$, any gain matrix for the PID gains could be chosen based on the desired response. This gain matrix was chosen such that the convergence rate about each axis of the inertial frame would be the same. The total desired torque input is then

$$T_d = T_{FL} + T_{PID} \tag{5.21}$$

Using this in eqn.(5.15) gives

$$\dot{h_T^B} = J_T \ddot{\Phi} = \hat{J}_T(K_i \int \Phi_e dt + K_p(\Phi_e) + K_d(\dot{\Phi}_e) + \ddot{\Phi}_d) \tag{5.22}$$

Assuming the estimate of the simulator inertia tensor to be accurate, eqn.(5.22) can be rewritten in the form

$$0 = K_i \int \Phi_e dt + K_p(\Phi_e) + K_d(\dot{\Phi}_e) + \ddot{\Phi}_e \tag{5.23}$$

using the definition

$$\ddot{\Phi}_d - \ddot{\Phi} = \ddot{\Phi}_e \tag{5.24}$$

Based on the characteristic polynomial of eqn.(5.23), a deadbeat controller may be designed. A deadbeat controller is a PID controller with gains chosen such that the resulting response has zero steady state error, minimum rise and settling times, less than 2% overshoot and undershoot, and very high control signal output [12]. The characteristic polynomial of eqn.(5.23) is

$$0 = K_i + K_p s + K_d s^2 + s^3 \tag{5.25}$$

The desired characteristic equation is

$$0 = 1 + \frac{\alpha_2 s}{w_n} + \frac{\alpha_1 s^2}{w_n^2} + \frac{s^3}{w_n^3} \tag{5.26}$$

where the constant α values are found by table look-up and ω_n is the closed loop natural frequency. From this requirement, the gains are found as

$$K_i = w_n^3 \tag{5.27}$$

$$K_p = w_n^2 \alpha_2 \tag{5.28}$$

$$K_d = w_n \alpha_1 \tag{5.29}$$

Therefore, based on the desired closed loop natural frequency, the gains K_d, K_p, K_i are defined. Note that eqn.(5.25) is based on a continuous controller. Since the sampling time of the controller is fast relative to the system dynamics, this approximation is justified for the discrete controller being used.

5.3 Quaternion Error

The IMU outputs the angular position of the simulator relative to the inertial frame. An expression for the angular position error between the desired and body frames is needed for the controller. Because Euler angles are limited by singularity points and perform poorly for large angle maneuvers, quaternions (Euler-Rodrigues symmetric parameters) were used to describe the state error. Instead of using three rotations as in Euler angles, quaternions represent an orientation as just one rotation. This development follows the explanation shown by Sidi [5]. A quaternion may be defined as

$$q = \begin{bmatrix} \hat{e}_x \sin(\frac{\theta}{2}) \\ \hat{e}_y \sin(\frac{\theta}{2}) \\ \hat{e}_z \sin(\frac{\theta}{2}) \\ cos(\frac{\theta}{2}) \end{bmatrix} \tag{5.30}$$

where θ is the magnitude of the rotation between frames and $\hat{e}_i$ are unit vectors describing the frame which the orientation is relative to. The first three elements of a quaternion can be interpreted as the vector portion $\overrightarrow{q}$, while the fourth element is the scalar portion q_4. A quaternion represents the rotation from one reference frame to another. The conjugate of the quaternion is defined as

$$q^* = \begin{bmatrix} -\hat{e}_x \sin(\frac{\theta}{2}) \\ -\hat{e}_y \sin(\frac{\theta}{2}) \\ -\hat{e}_z \sin(\frac{\theta}{2}) \\ cos(\frac{\theta}{2}) \end{bmatrix} \tag{5.31}$$

The conjugate of a quaternion represents the opposite orientation or rotation. The result of the quaternion multiply operation between two quaternions can be interpreted as the rotation between the two sets of axes. As such, the rotation of the current position and to the desired position can be written as

$$q'' = q \otimes q' \tag{5.32}$$

The quaternion multiply function may alternatively be implemented in matrix multiplication form as

$$q'' = q \otimes q' = \begin{bmatrix} q'_4 & q'_3 & -q'_2 & q'_1 \\ -q'_3 & q'_4 & q'_1 & q'_2 \\ q'_2 & -q'_1 & q'_4 & q'_3 \\ -q'_1 & -q'_2 & -q'_3 & q'_4 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{5.33}$$

The objective is to find an expression for the angular position error. The error is defined as the rotation from the simulator frame into the desired frame. Equation (5.34) shows how this quaternion error is found.

$$q_s^* \otimes q_d = q_e \tag{5.34}$$

This equation may be interpreted as meaning the orientation of the inertial frame in the simulator frame quaternion multiplied with the orientation of the desired frame in the inertial frame results in the orientation of the desired frame relative to the body frame.

A useful trigonometric identity is

$$\sin(\theta/2)\cos(\theta/2) = \frac{1}{2}\sin(\theta) \tag{5.35}$$

From this, and based on eqn.(5.30) the position error can be unwrapped as

$$\Phi_e = 2\overrightarrow{q_e}q_{e4} = 2\hat{e}\sin(\theta/2)\cos(\theta/2) = \hat{e}\sin\theta \tag{5.36}$$

This result shows the significance of the quaternion error. From this definition, it follows that

$$\int \Phi_e dt = \int 2\overrightarrow{q_e}q_{e4}dt \tag{5.37}$$

At this pont, Φ can be more concisely described as

$$\Phi = 2\overrightarrow{q_s}q_{s4} \tag{5.38}$$

Because the angular velocity output of the IMU is in terms of the body frame, the angular velocity error term did not need be transformed. The error of angular velocity is simply

$$\dot{\Phi}_e = \dot{\Phi}_d - \dot{\Phi} \tag{5.39}$$

and the acceleration error is

$$\ddot{\Phi}_e = \ddot{\Phi}_d - \ddot{\Phi} \tag{5.40}$$

5.4 Trajectory Generation

Trajectory generation determines suitable Φ_d, $\dot{\Phi}_d$, and $\ddot{\Phi}_d$ values based on the desired final positions. A list of desired final angular positions and times are read in as a .txt file. Based on these desired final positions, the trajectory generation provides the values of position and its derivatives as a function of time.

The trapezoidal trajectory being developed is limited by a specified constant maximum velocity (V_{max}) and acceleration (A_{max}). Figure (5.1) shows an example of a trapezoidal trajectory.

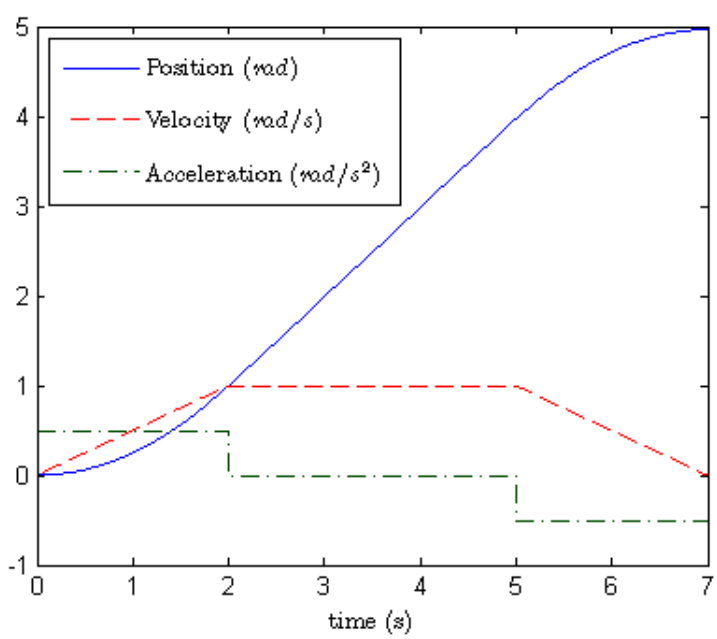


Fig. 5.1: Example of a trapezoidal trajectory

In this figure, the maximum acceleration is in effect for the first two seconds. The next three seconds are at V_{max} and no acceleration occurs. From five to seven seconds, the maximum deceleration is in effect. Based on the magnitude of the position error and A_{max}, the trajectory generator may never even reach V_{max}.

The desired position needs to be reformed into a quaternion in order to find the quaternion error. This quaternion trajectory is shown in eqn. (5.41).

$$q_{traj} = \begin{bmatrix} \hat{e}_x \sin(\theta_d/2) \\ \hat{e}_y \sin(\theta_d/2) \\ \hat{e}_z \sin(\theta_d/2) \\ \cos(\theta_d/2) \end{bmatrix} \tag{5.41}$$

The unit vector $\hat{e}$ and θ_d of this quaternion have been extracted from the quaternion $q_s^* \otimes q_{reference}$. The trapezoidal trajectory is generated relative to the start position of the trajectory. Because of this, a quaternion multiply can be used to find the desired quaternion as shown in eqn.(5.42).

$$q_d = q_{traj}^* \otimes q_{start} \tag{5.42}$$

where q_{start} is the simulator quaternion q_s at the beginning of the maneuver.

5.5 External Motor Controller

In addition to the internal servo controller of the motor, another controller was designed for the dynamics of the motor. The servo controller requires a rate command instead of a torque command in the closed loop operating mode. Based on the motor model, the wheel velocity for each wheel can be found in terms of the desired toque as shown in eqn.(5.43).

$$\dot{\Omega} = \frac{1}{\tau}(\Omega_{cmd} - \Omega) \tag{5.43}$$

where Ω_{cmd} is the command which reaches the servo controller. Taking the Laplace transform of eqn. (5.43) and rearranging leads to the transfer function

$$\frac{\dot{\Omega}}{\Omega_{cmd}} = \frac{s}{\tau s + 1} \tag{5.44}$$

The external servo controller shown in fig.(5.45) was designed with the purpose of canceling the motor dynamics.

$$\frac{\Omega_{cmd}}{\Omega_d} = \frac{\hat{\tau} s + 1}{s} = \hat{\tau} + \frac{1}{s} \tag{5.45}$$

Note that Ω_d is actually the desired angular acceleration for this choice of controller. Thus, using this external servo controller before the motor will result in the wheels reaching the desired angular acceleration based on the motor model. This means that only multiplying by the desired acceleration and an integration of the desired acceleration are required to compensate for the motor dynamics. This integration operation was performed discretely using trapezoidal integration. The motor model does not take into account any nonlinear affects of the motor. The resolution error of motor rates may cause significant error when using trapezoidal integration and degrade the external servo controller performance. The motor reaching the maximum torque it can provide may also limit this controller performance.

5.6 Controller Implementation

The output of the IMU and the Hall sensors are used by the controller to determine the desired wheel velocities Ω_{cmd}. Figure (5.2) illustrates the structure of the controller with respect to the sensor outputs.

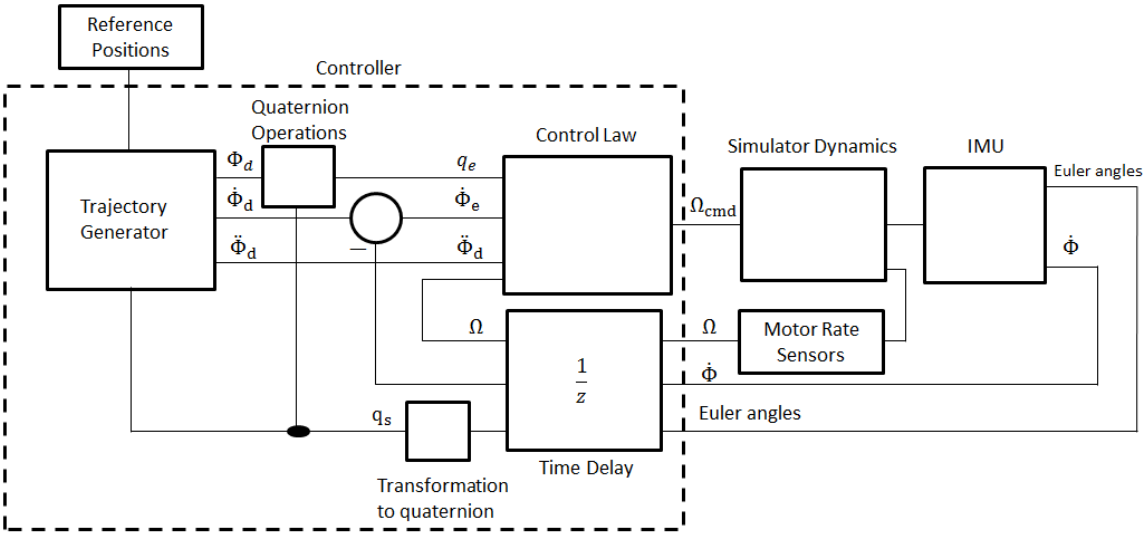


Fig. 5.2: Closed loop system structure

The reference is the input specified by the user in the form of a text file which is read in at runtime. The references are quaternion positions and times to begin approaching the positions. The Euler angles output by the IMU are transformed into a quaternion by the controller. The resulting quaternion is used by the trajectory generator once per reference to determine the desired position trajectory to reach the reference position based on the current position. The simulator quaternion and the desired position are used at each time step to determine the position error.

5.7 Linear System Approximation

Under the control law with feedback linearization, the system can be simplified. The state space representation of the uncontrolled linearized system treated as a continuous

system is given by eqns.(5.46-5.51).

$$x = \begin{bmatrix} \dot{\Phi} \\ \Phi \\ \int \Phi dt \\ \Omega \end{bmatrix} \tag{5.46}$$

$$\dot{x} = Ax + B\Omega_d \tag{5.47}$$

$$y = x \tag{5.48}$$

$$\tag{5.49}$$

$$A = \begin{bmatrix} [0] & [0] & [0] & -J_s^{-1} J_w^w G/\tau \\ [I] & [0] & [0] & [0] \\ [0] & [I] & [0] & [0] \\ [0] & [0] & [0] & -\frac{1}{\tau}[I] \end{bmatrix} \tag{5.50}$$

$$B = \begin{bmatrix} J_s^{-1} J_w^w G/\tau \\ [0] \\ [0] \\ \frac{1}{\tau}[I] \end{bmatrix} \tag{5.51}$$

This model of the plant neglects any amount of nonlinearities which are not canceled by feedback linearization. Also, the wheel velocities required for feedback linearization are not accounted for by this model. A classical representation of this plant with a PID controller is shown in fig.(5.3).

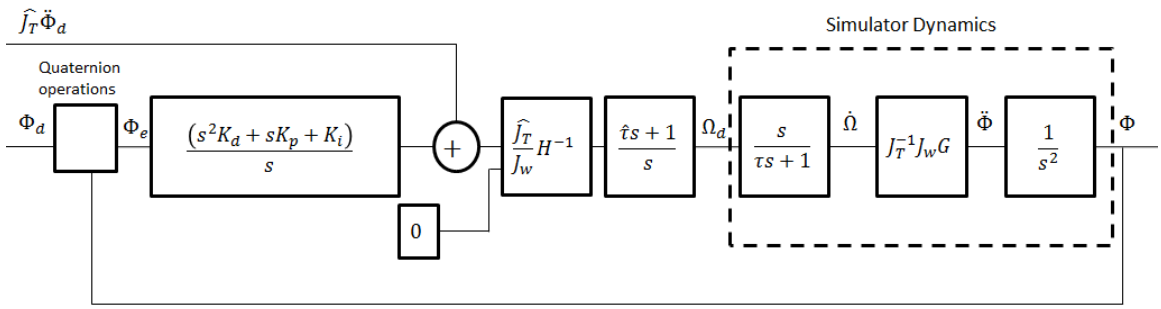


Fig. 5.3: Continuous linear approximation of controlled system dynamics

5.8 Nonlinear System Approximation

The nonlinear system model shown in fig. (5.4) is based on eqn.(5.15). This figure shows the nonlinear system dynamics with a controller that implements PID and feedback linearization control.

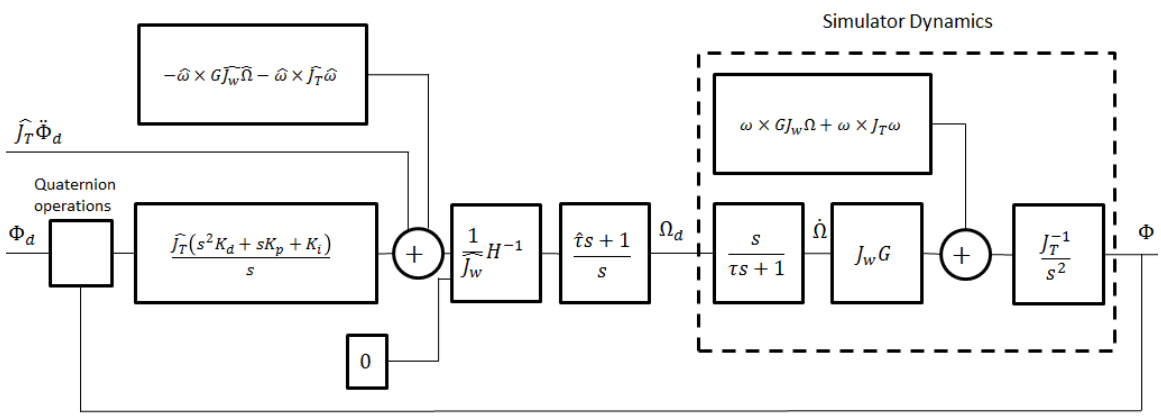


Fig. 5.4: Continuous nonlinear approximation of controlled system dynamics

Note that if the desired angular velocity and its derivative are zero, the closed loop transfer function is

$$\frac{\Phi}{\Phi_d} = \frac{sK_p + K_i}{s^3 + K_d s^2 + sK_p + K_i} \tag{5.52}$$

The analytic response resulting for this case can be used to compare with the physical system response.

If $\int \Phi_d dt$, Φ_d, $\dot{\Phi}_d$, $\ddot{\Phi}_d$ are all specified the closed loop transfer function is

$$\frac{\Phi}{\Phi_d} = \frac{s^3 + K_d s^2 + sK_p + K_i}{s^3 + K_d s^2 + sK_p + K_i} = 1 \tag{5.53}$$

This means that the simulator should exactly track the desired trajectory based on this model. This model is only an approximation of the system dynamics however and the simulator will not exactly track the desired trajectory in actuality.

Chapter 6

Testing

6.1 System Properties

All testing results use the data presented unless otherwise stated.

Table 6.1: Table of testing properties

Property	Value
Wheel inertia J_w	$1.53e^{-4}kgm^2$
Expected motor time constant $\hat{\tau}$	$0.0s$
Motor filter cutoff frequency	$20Hz$
External gyro rate filter cutoff frequency	$20Hz$
Sampling period	$20ms$

The moment of inertia of the simulator and reaction wheels is

$$J_T = \begin{bmatrix} 2.99e^{-2} & -1.04e^{-5} & 9.23e^{-5} \\ -1.04e^{-5} & 3.03e^{-2} & 1.84e^{-6} \\ 9.23e^{-5} & 1.84e^{-6} & 4.66e^{-2} \end{bmatrix} kgm^2 \tag{6.1}$$

This moment of inertia was estimated through a computer aided drafting software. Since some hardware changes to the simulator have been made and not accounted for, this is a very approximate estimate.

6.2 IMU Performance

For this test, the simulator was set on a stand so it could not move, even if the wheels were set to run. The gyros were calibrated using internal IMU software before this test. The goal of this test was to evaluate the performance of the IMU. Figure (6.1) and fig.(6.2) show the angular position and velocity respectively for a test where the table was stationary and the motors were not commanded to run.

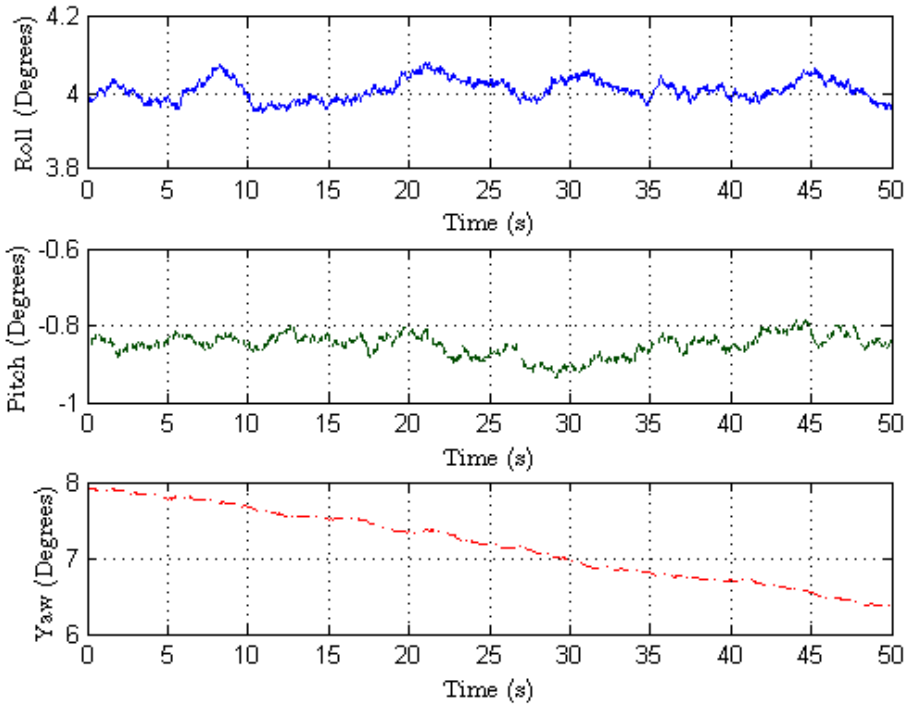


Fig. 6.1: Euler angles of a fixed simulator test

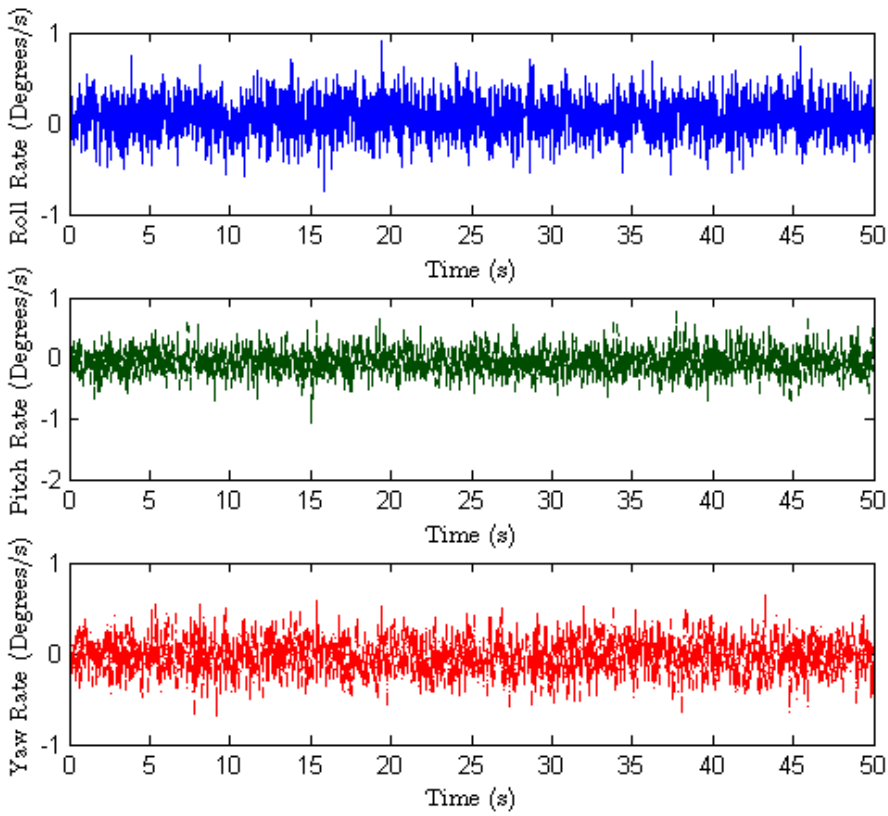


Fig. 6.2: Angular velocity of a fixed simulator test

These figures show the output from the IMU; no external filtering is shown. While the gyro rate sensors stayed close to zero, the position in yaw continued to drift. The drift rate of yaw is $1.9 deg./min.$. Based on the data in fig.(6.1) and fig.(6.2), table (6.2) shows the standard deviation and mean of the measured variables.

Table 6.2: Table of mean and standard deviation for drift test

	Mean	Std. dev.
Roll	$4.01 deg.$	$2.79e^{-2} deg.$
Pitch	$-8.50e^{-1} deg.$	$2.86e^{-2} deg.$
$\dot{\Phi}_x$	$6.64e^{-2} deg.$	$2.14e^{-1} deg.$
$\dot{\Phi}_y$	$-6.01e^{-2} deg.$	$2.17e^{-1} deg.$
$\dot{\Phi}_z$	$2.68e^{-2} deg.$	$1.99e^{-1} deg.$

The mean of the gyro rates is not zero. Based on testing before and after gyro calibration, it can be inferred that the majority of the drift is due to the fact that the IMU uses the gyro readings to get a better estimate of position. Gyro bias error effectively causes the estimate of position to drift.

Additional error may be a result of the magnetic fields of electronics interfering with the magnetometer. Because motors can produce magnetic fields of significant strength, tests with the motor given a command were performed. However, these tests had similar results to tests without a command to the motor. While these tests lasted only 50 seconds, it is possible that affects of the change in temperature on the sensors may be seen after a longer amount of time. This source of error is likely minimal since the IMU uses temperature sensors to compensate for this effect.

6.3 Motor Response

The dynamics of the motors can be significant in system control. Because of this, a dynamic model of the motor including the servo controller was needed. For this test, the simulator was again set on a stand so the simulator would not move. The speed of a motor was commanded to reach a speed of $200 rad/s$. Figure (6.3) shows the response of the motor given this step input and the expected response using the developed motor model.

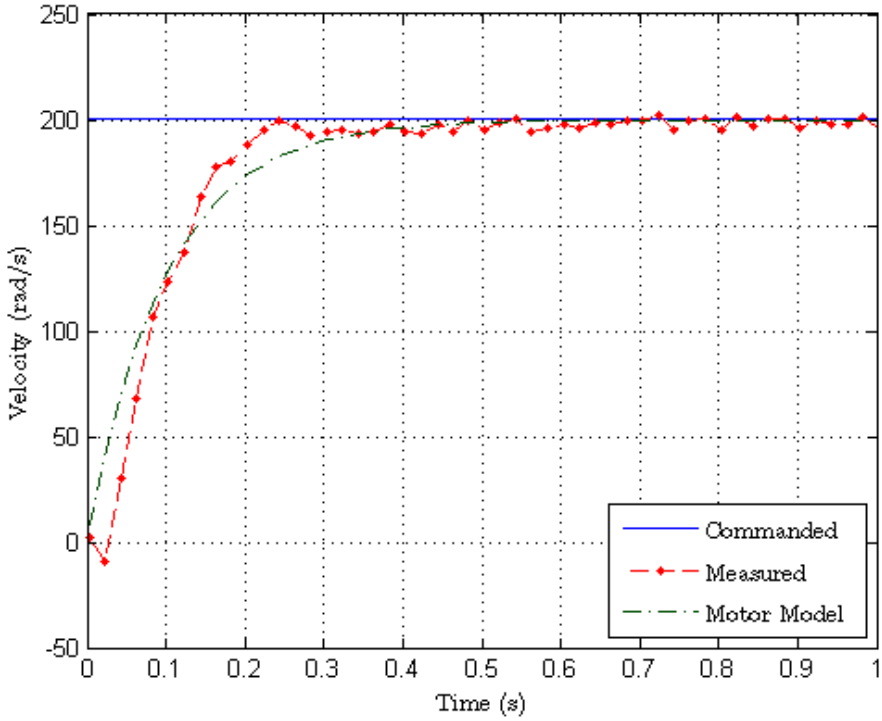


Fig. 6.3: Unfiltered motor response with step input

This response shows the motor model developed tracks the actual response. The motor transfer function was approximated as being first order with a time constant of $0.1s$ and a unity gain.

During testing it became apparent that the cancellation of the motor dynamics by the external motor controller caused significant steady state oscillations. For a closer look at these oscillations, consider fig.(6.4) and fig.(6.5).

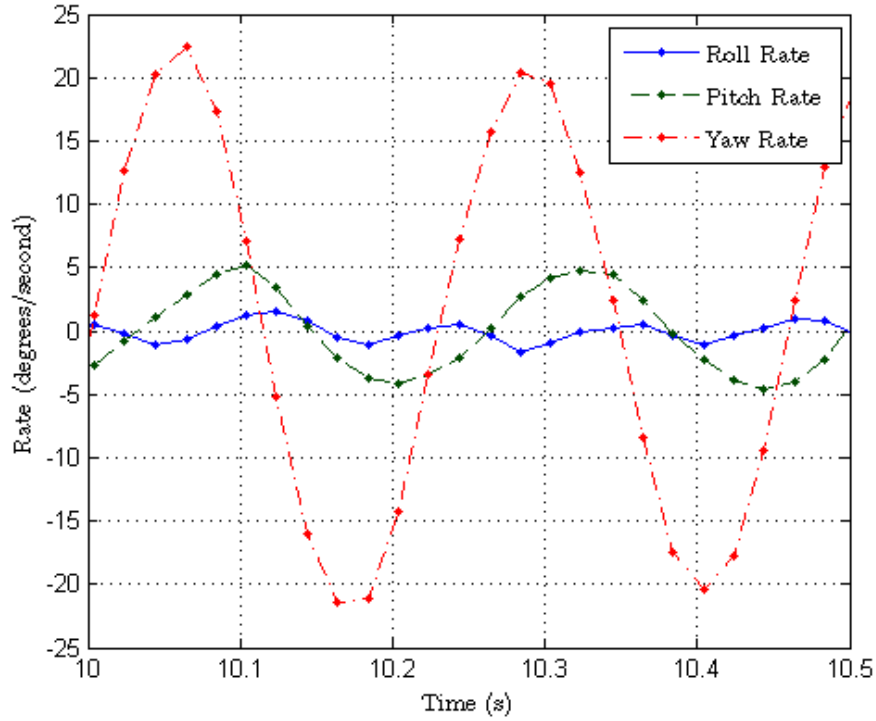


Fig. 6.4: Angular velocity of the simulator using $\hat{\tau} = 0.1$

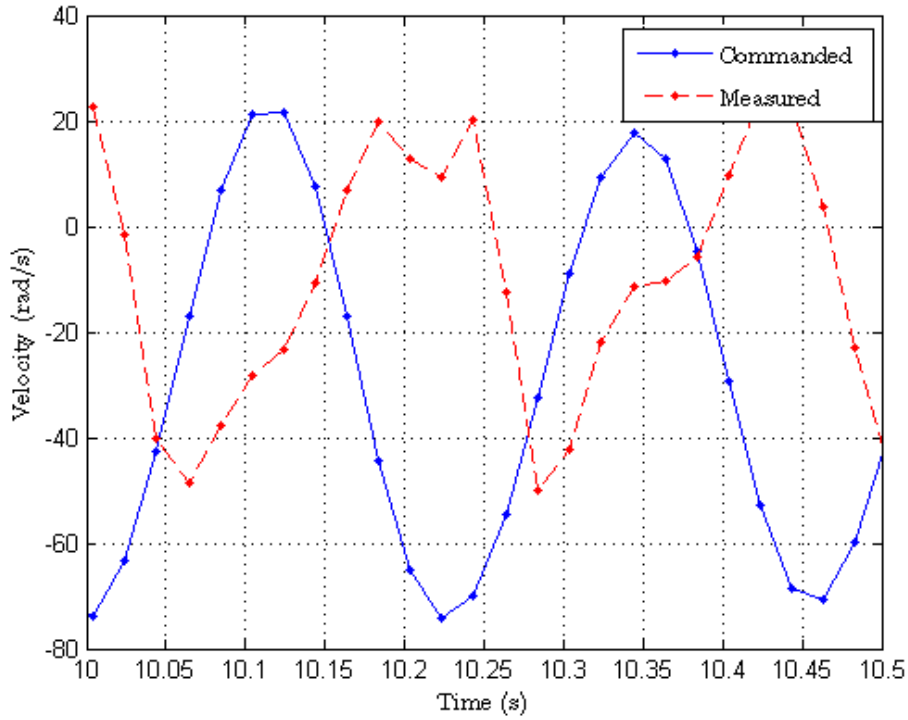


Fig. 6.5: Velocity of wheel 0 using $\hat{\tau} = 0.1$

These figures are taken from the same test a stationkeeping test using $\hat{\tau} = 0.1$. The angular velocity of the simulator oscillates with a frequency of about $4Hz$. Figure (6.5) shows the oscillation of the momentum wheels having a frequency of about $4Hz$ with about the same phase as the negative angular rates. The angular rates are 90° out of phase with the angular acceleration of the simulator. The wheel speeds are 90° out of phase with the wheel torques. Thus, the wheels attempt to counter the higher angular acceleration. From this it can be concluded that the oscillations are not effectively dampened by the controller. This could be attributed to the expected motor time constant $\hat{\tau}$ being too small compared to the sampling time. Latency of the Hall sensors could be the cause. It is also possible that dynamics of the servo motor controller which weren't accounted for affect the response. Because of these oscillations, $\hat{\tau} = 0$ was used for testing.

6.4 Stationkeeping

The simulator was set on the air bearing and free to move. Based on the way the simulator oscillated about its center of rotation, the simulator table was balanced by tightening the balancing screws. A commanding computer was attached to the BB through an ethernet cable. A command to execute the code after an amount of time was given and the commanding computer was detached. For safety, a maximum wheel speed was set by the controller code.

For the first stationkeeping test, a controller natural frequency of $1rad/s$ was selected. Only one reference position was used so that, after the initial response, the steady state behavior could be observed. Figure (6.6) shows the position of the simulator quickly converging to the reference position, having a period of accurate pointing, and finally succumbing to momentum saturation. The reason for the divergence in position near the final time is that the wheels are reaching their maximum allowed speeds near $110s$. Figure (6.7) shows the commanded and measured wheel velocities. Near $110s$ it is clear that the wheel is not reaching the desired velocities but only reaching the maximum allowed velocity.

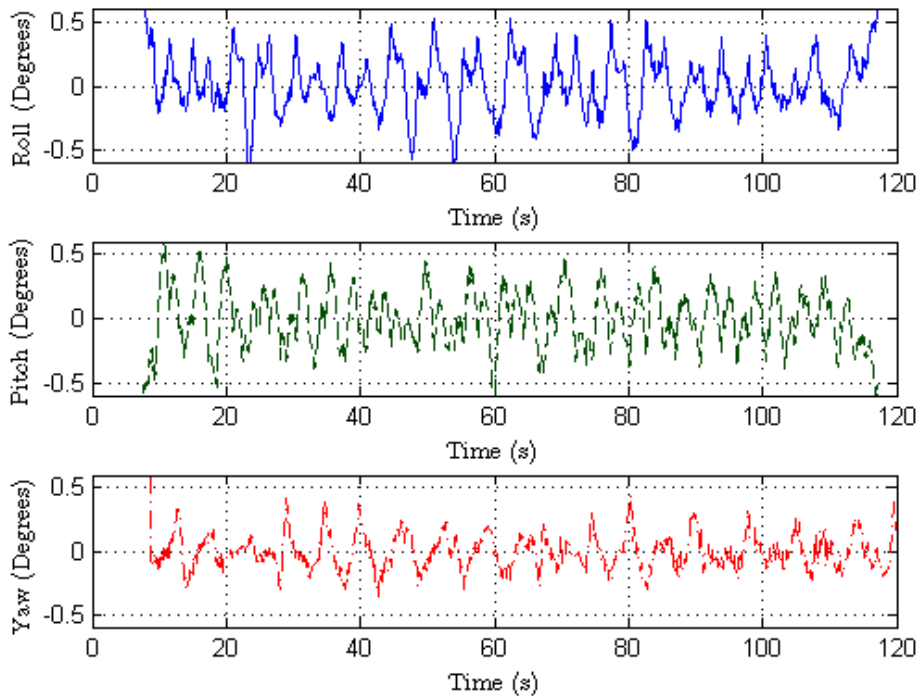


Fig. 6.6: Euler angles of the simulator for stationkeeping test with $\omega_n = 1 rad/s$

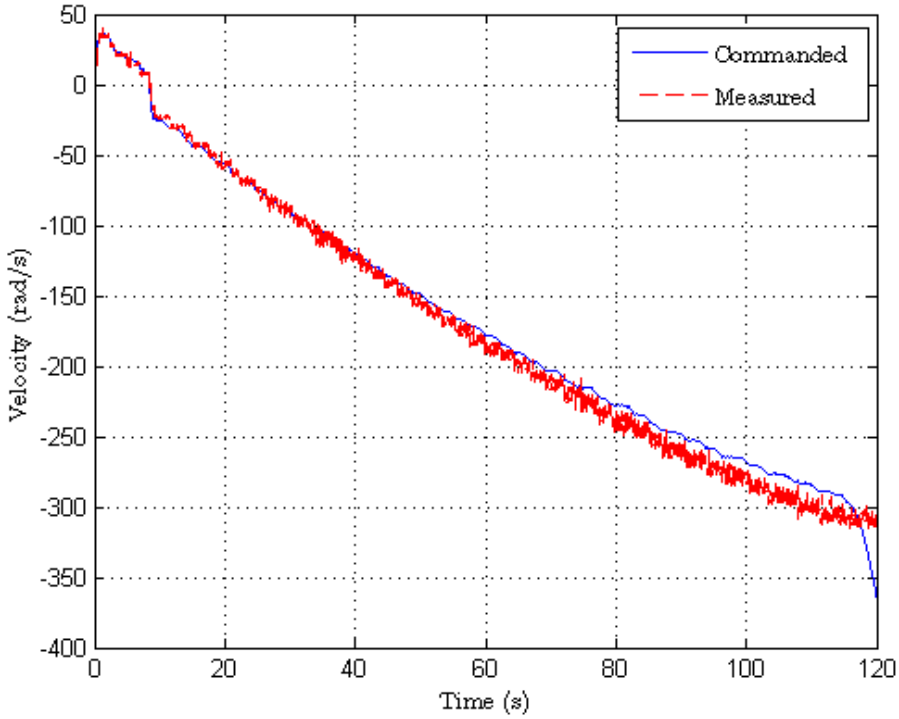


Fig. 6.7: Velocity of wheel 0 for stationkeeping test with $\omega_n = 1rad/s$

Table (6.3) shows the mean and standard deviation of angular position for data from $20 - 110s$. Note that the Euler angles and Φ_e values are nearly the same but with an opposite sign for this test which has only small angle error.

Table 6.3: Table of mean and standard deviation for stationkeeping test with $\omega_n = 1rad/s$

	Mean	Std. dev.
Roll	$-1.07e^{-3}deg.$	$2.20e^{-1}deg.$
Pitch	$1.18e^{-2}deg.$	$1.96e^{-1}deg.$
Yaw	$-2.53e^{-3}deg.$	$1.32e^{-1}deg.$
Φ_{ex}	$1.07e^{-3}deg.$	$2.20e^{-1}deg.$
Φ_{ey}	$-1.17e^{-2}deg.$	$1.96e^{-1}deg.$
Φ_{ez}	$2.51e^{-3}deg.$	$1.32e^{-1}deg.$

The L_2 norm of the mean Φ_e components is $1.20e^{-2}deg.$. This norm represents the total angular error.

A second stationkeeping test is presented using a controller natural frequency of $2rad/s$. As shown in fig.(6.8) and fig.(6.9), the motors take about $110s$ to reach saturation speeds.

Other motors never reach saturation during this test. It is possible to use momentum dumping to solve the problem of saturation but this is not used in this project.

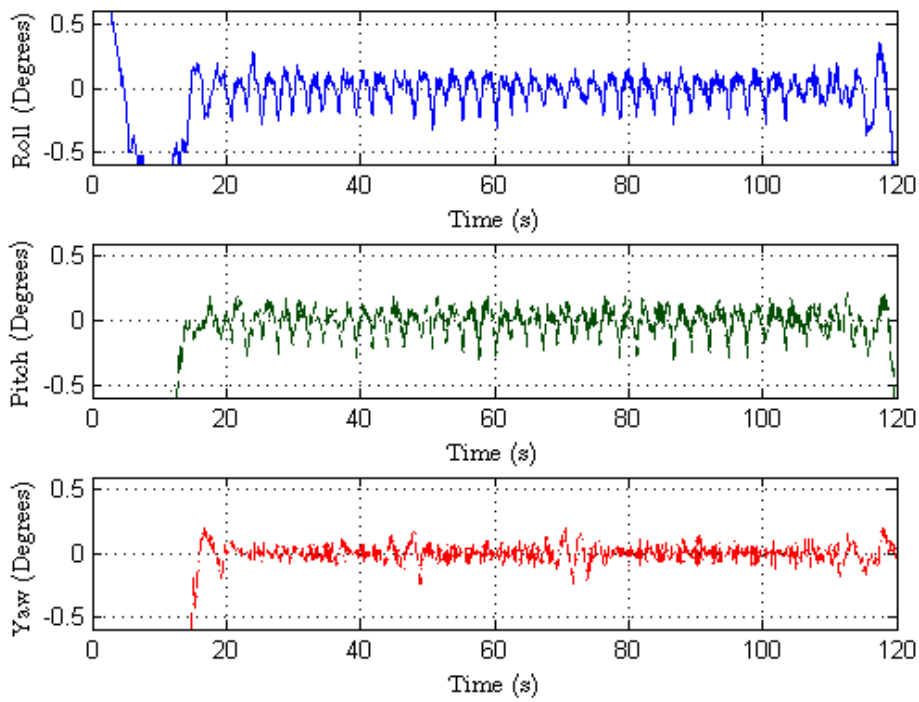


Fig. 6.8: Euler angles of the simulator for stationkeeping test with $\omega_n = 2rad/s$

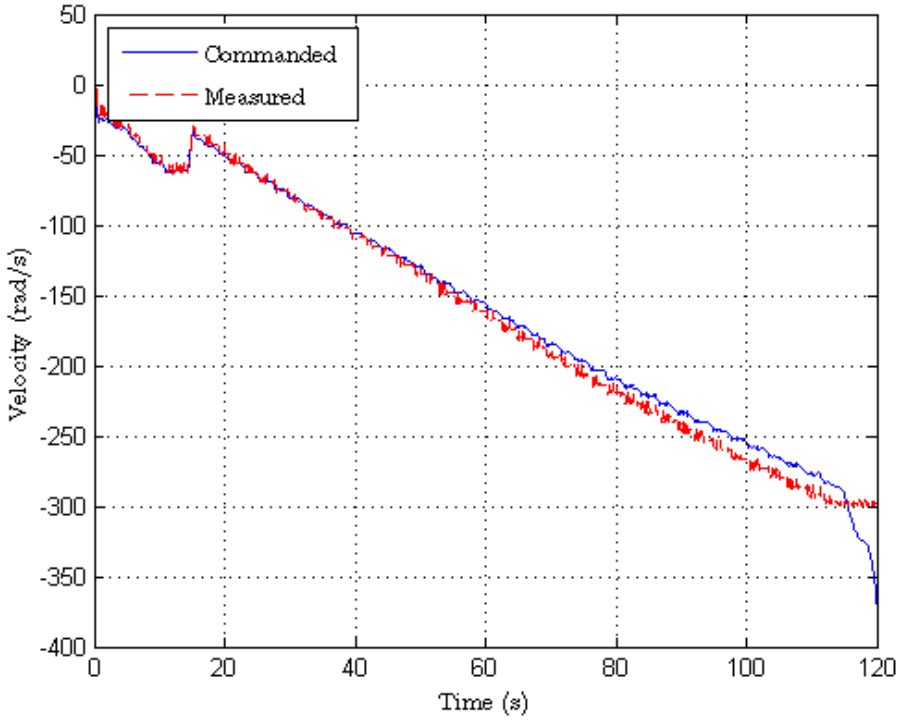


Fig. 6.9: Velocity of wheel 2 for stationkeeping test with $\omega_n = 2rad/s$

Table (6.4) shows the mean and standard deviation of angular position of the simulator for $20 - 110s$.

Table 6.4: Table of mean and standard deviation for stationkeeping test with $\omega_n = 2rad/s$

	Mean	Std. dev.
Roll	$-1.20e^{-3}deg.$	$9.53e^{-2}deg.$
Pitch	$-2.73e^{-4}deg.$	$9.23e^{-2}deg.$
Yaw	$9.37e^{-5}deg.$	$5.36e^{-2}deg.$
Φ_{ex}	$1.20e^{-3}deg.$	$9.53e^{-2}deg.$
Φ_{ey}	$2.77e^{-4}deg.$	$9.32e^{-2}deg.$
Φ_{ez}	$-4.72e^{-5}deg.$	$5.36e^{-2}deg.$

The L_2 norm of the mean Φ_e components is $1.23e^{-3}deg.$. Compared to the test using $\omega_n = 1rad/s$, this norm is almost an order of magnitude smaller. The standard deviations of the $\omega_n = 1rad/s$ test are also larger. Clearly, using this higher natural frequency in the controller improves the performance of the system.

6.5 Control of Motion in One Degree of Freedom

The controller was used to move the simulator to the desired position in yaw while setting the desired position in roll and pitch to zero. This was done in order to focus on the performance of the controller in one dimension. Similarly to the stationkeeping test, the simulator was set on the air bearing and free to move. A commanding computer was attached to the BB through an ethernet cable. A command to execute the code after an amount of time was given and the commanding computer was detached. The reference positions and times were input by the user. From this reference, a trajectory is generated by the program in real time.

6.5.1 One Degree of Freedom Test without Trajectory Generation

As another test, reference points proceed around in yaw in 90° increments without trajectory generation. If instead the references were to use 180° increments, the controller may create a trajectory which is physically impossible for the simulator since the simulator has ±45° of freedom in pitch and roll in the inertial frame. For this test, the value $\omega_n = 2$ was used. Figure (6.10) shows the roll, pitch and yaw for this test. Note that there is 48% overshoot at a peak time of $1.46s$ which is close to the 42% overshoot at a peak time of $1.35s$ predicted by eqn.(5.52). This comparison using Euler angles gives only an approximation of the overshoot and peak time.

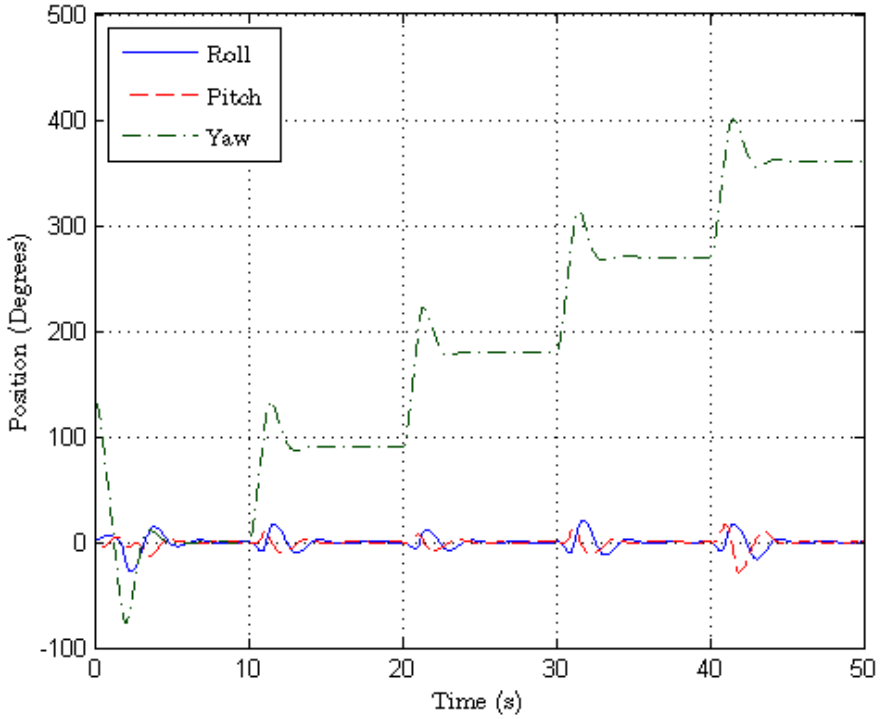


Fig. 6.10: Euler angles of the simulator with $\omega_n = 2rad/s$ for the no trajectory test

Figure (6.11) shows the position in yaw relative to the reference value. While there is some overshoot, the yaw converges on the reference position.

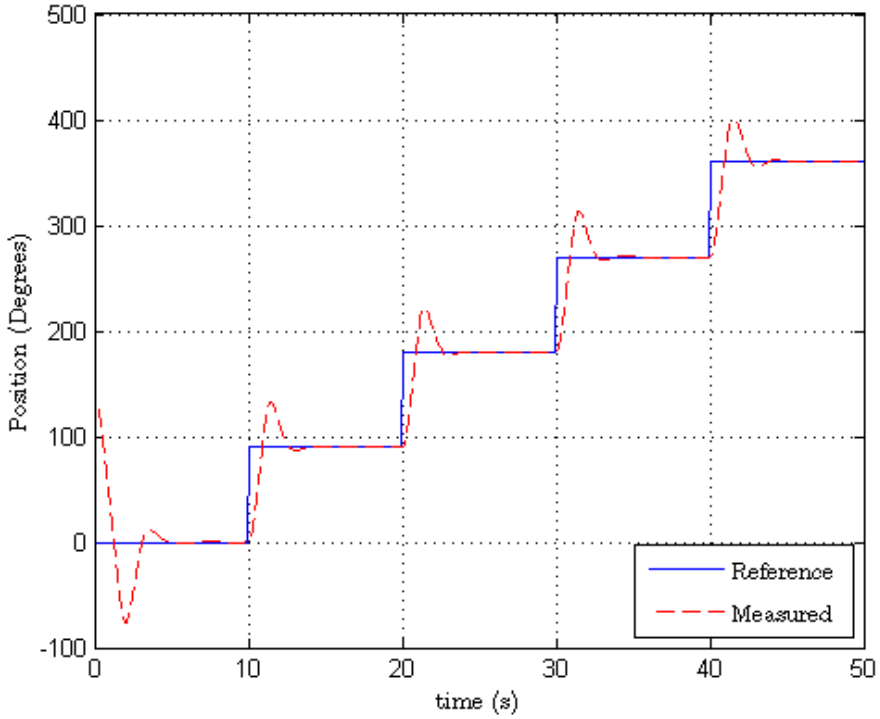


Fig. 6.11: References and measured position in yaw with $\omega_n = 2rad/s$ for the no trajectory test

The wheels do not reach the maximum speed for significant periods of time in this test as shown in fig.(6.12). When the yaw is settling to a new value, the commanded wheel velocity is typically larger than the maximum allowed wheel speed. However, this only happens briefly and the simulator still reaches the reference value.

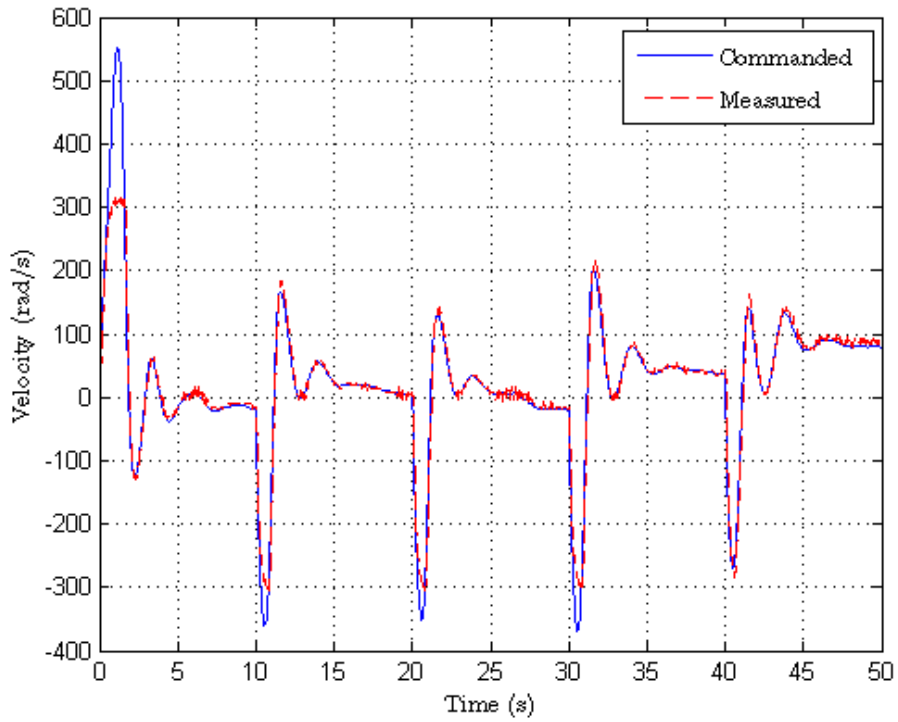


Fig. 6.12: Velocity of wheel 0 with $\omega_n = 2rad/s$ for the no trajectory test

6.5.2 One Degree of Freedom Test with Trajectory Generation

Another test was performed similarly to the previous test however, a trajectory was used as the desired states. The deadbeat controller had a natural frequency of $2rad/s$ for this test. Figure (6.13) shows the yaw against the pitch and roll.

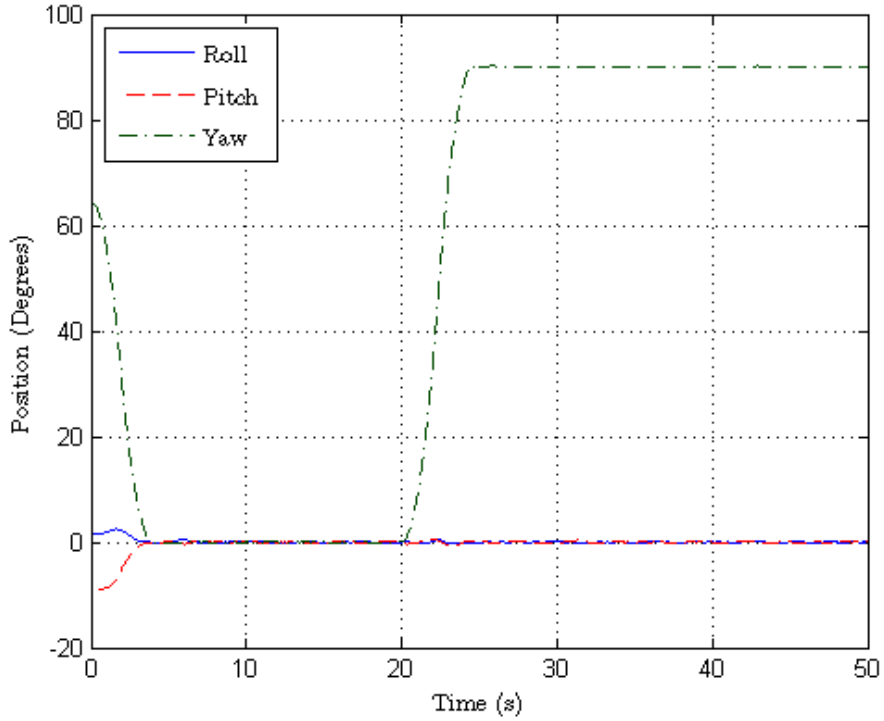


Fig. 6.13: Euler angles showing control in yaw using trajectory generation

Figure (6.14) is a plot of the trajectory, reference and measured position in yaw. From this figure, it is apparent the position in yaw is controlled and follows the trajectory.

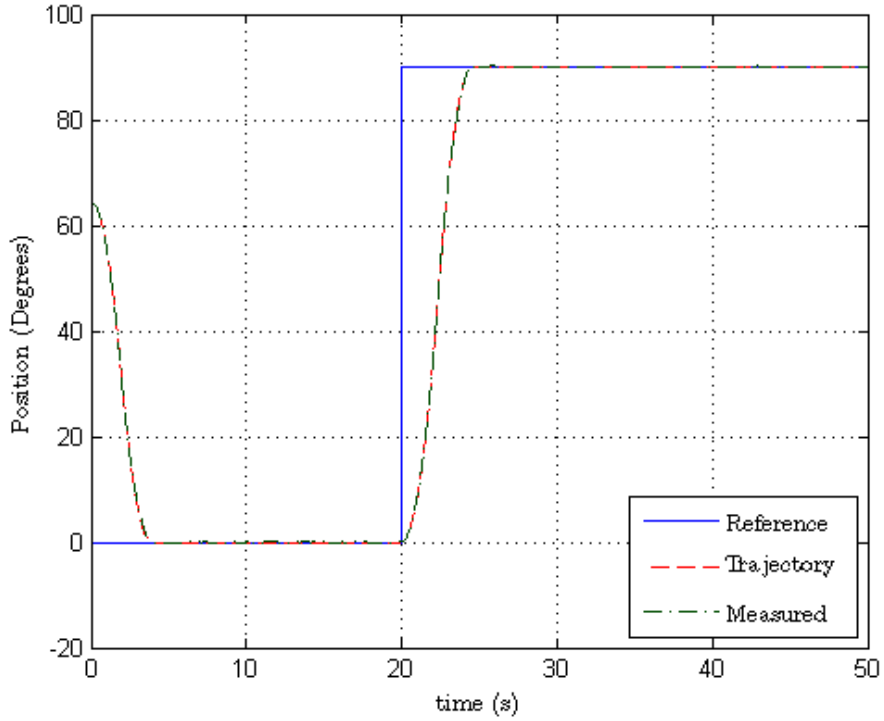


Fig. 6.14: Reference, trajectory, and measured angular position in yaw using trajectory generation

6.6 Three Degree of Freedom Response with Trajectory

Using the trapezoidal trajectory, the desired angular position, velocity, and acceleration were found based on the desired angular position input by the user. This test uses all three degrees of freedom and multiple reference points to evaluate the ability to track the desired trajectory.

Figures (6.15-6.17) show the measured position, the trajectory, and the input reference positions in roll, pitch, and yaw using $\omega_n = 1$. These figures show that the system tracks the trajectory in all three degrees of freedom. The position error is probably due to noise and external torque. The position in yaw has less error than that in roll and pitch. This is because the center of gravity is not aligned with the center of rotation in the z body axis. The resulting external torque acts to keep the simulator at zero roll and pitch.

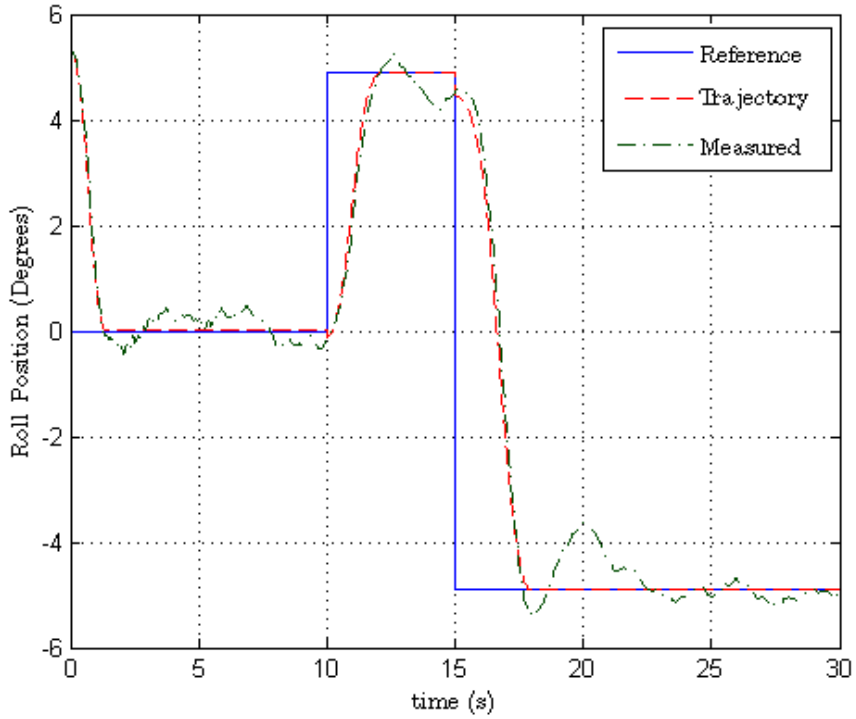


Fig. 6.15: References, trajectory, and measured position in roll with $\omega_n = 1rad/s$ for three degree of freedom test

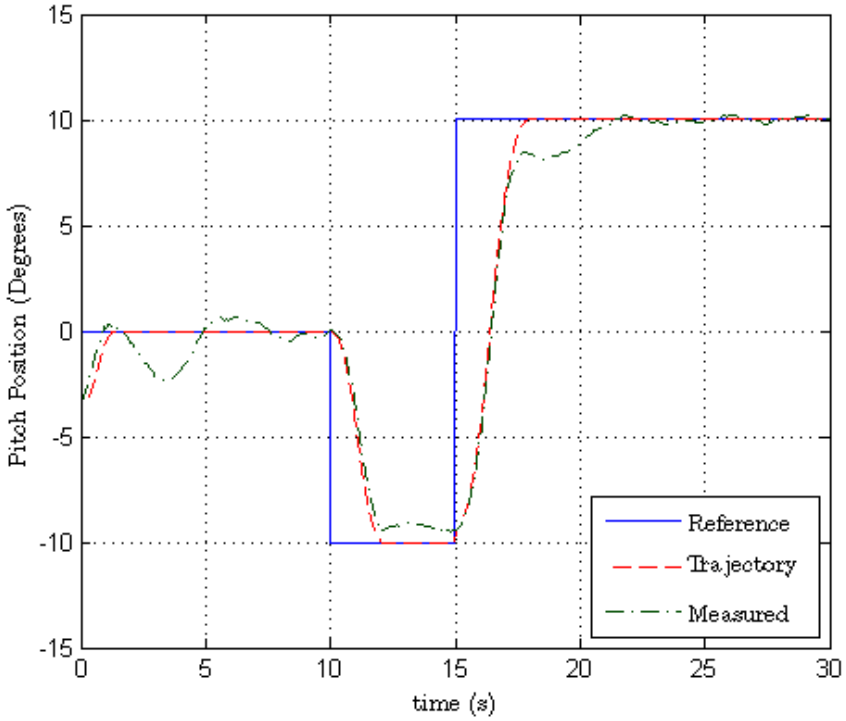


Fig. 6.16: References, trajectory, and measured position in pitch with $\omega_n = 1 rad/s$ for three degree of freedom test

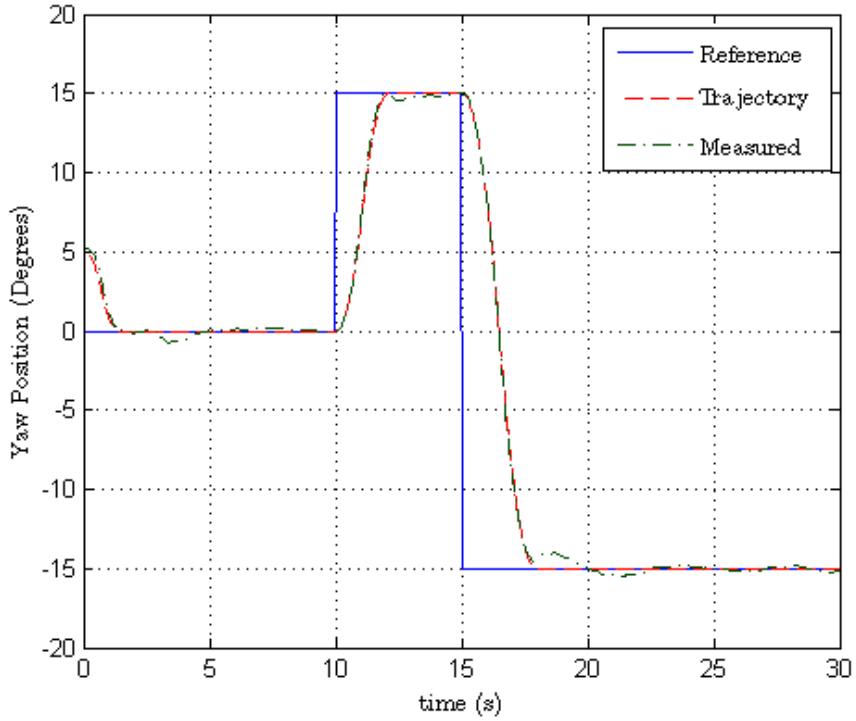


Fig. 6.17: References, trajectory, and measured position in yaw with $\omega_n = 1rad/s$ for three degree of freedom test

Figure (6.18) shows the wheel velocity growing large when holding the last attitude. This is because the external torque requires the wheels to continually increase in velocity in order to maintain a constant position. It is very difficult to align the center of gravity with the center of rotation accurately in all directions simultaneously as described by Fullmer [13]. Because this project focused on pointing in yaw, balancing about the x and y body axes was the primary concern.

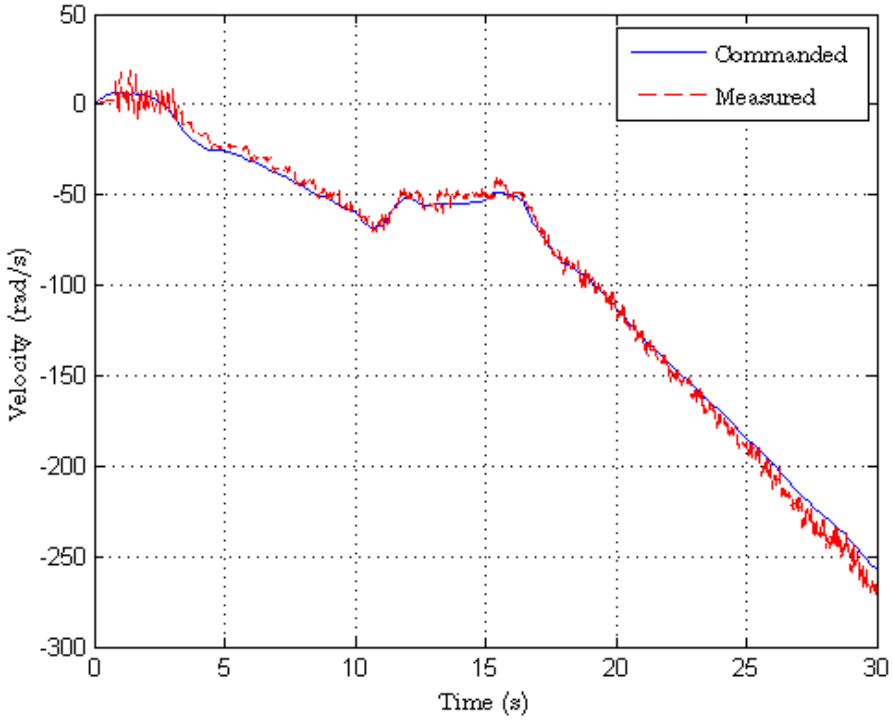


Fig. 6.18: Velocity of wheel 0 with $\omega_n = 1rad/s$ for three degree of freedom test

This test was repeated for $\omega_n = 2rad/s$. Figures (6.19-6.21) display the results of this test. Notice that the trajectory in roll does not directly converge to 0°. This is because the trajectory is defined through quaternions, and not the Euler angles which are shown.

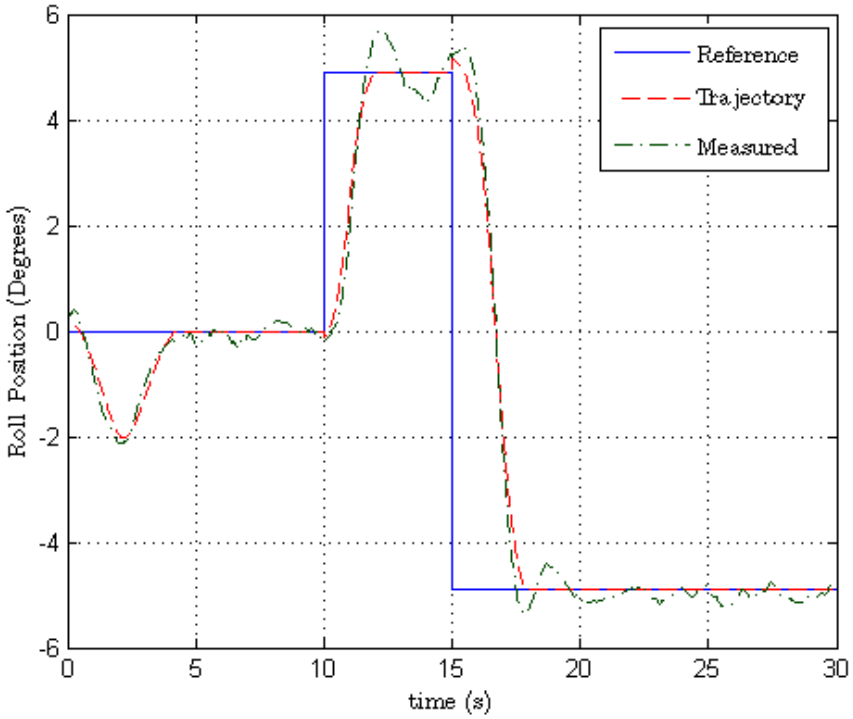


Fig. 6.19: References, trajectory, and measured position in roll with $\omega_n = 2rad/s$ for three degree of freedom test

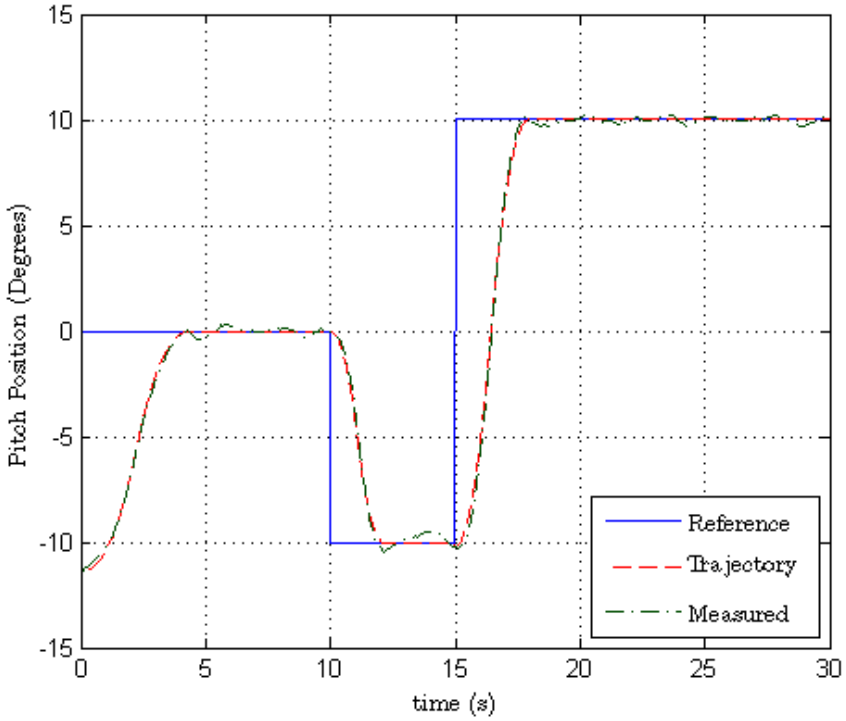


Fig. 6.20: References, trajectory, and measured position in pitch with $\omega_n = 2rad/s$ for three degree of freedom test

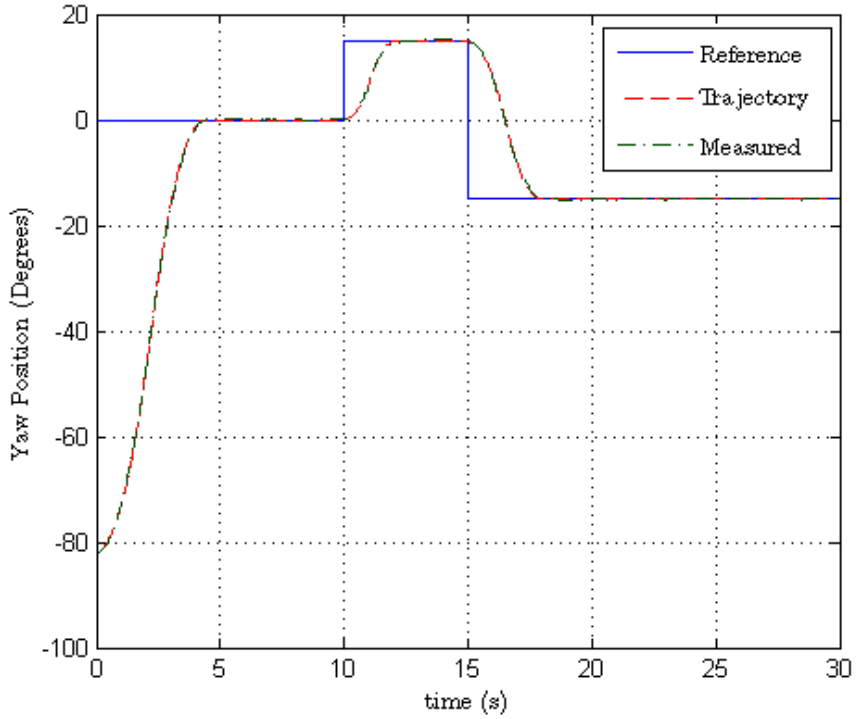


Fig. 6.21: References, trajectory, and measured position in yaw with $\omega_n = 2rad/s$ for three degree of freedom test

Chapter 7

Conclusions and Future Work

This project demonstrates the implementation of reaction wheels for attitude control. A system model was developed and used to design a controller. This controller used trajectory generation and deadbeat tuning for the purpose of improving system performance. The motor and IMU performance was tested and discussed. One and three degree of freedom test results were investigated and trajectory generation has been demonstrated. The interaction between the controller and the wheel saturation speed have been discussed. Based on the results, it is clear that the controller moves the simulator to the desired position. Limitations of the closed loop system were discussed. The controller boasted control in all three degrees of freedom and had pointing accuracy within $0.4°$ in some cases.

The magnetometer estimate of north was significantly inaccurate. This may be because the magnetometer needs to be calibrated again or a IMU option for North compensation needs to be adjusted. Soft and hard iron calibration is needed to compensate for ferromagnetic components that affect the magnetometer.

The gyro bias which may lead to an inaccurate estimate of North may be calibrated for. Since the IMU uses gyro measurements to improve angular position estimates, decreasing the gyro bias will improve the position estimates.

A better estimate of the simulator inertia tensor would help decrease both overshoot and errors which come from feedback linearization. The inertia tensor used was taken from a CAD model. The CAD model does not accurately represent the inertia since many hardware changes have been made.

A momentum dumping scheme would greatly improve the performance of the simulator by preventing the wheels from approaching the maximum speed. Momentum dumping is accomplished by using the interaction between the wheels to slow each other. This would

be implemented as part of the controller.

The resolution error of the wheel velocity makes the system performance significantly worse. However, there are many ways to fix this problem. One fix would be to use the servo controller's digital input/output instead of its analog input/output. This solution would require some program code to be rewritten. Another way to avoid this problem is to use smaller reaction wheels. This however, would lead to earlier wheel speed saturation.

The response of the motor with servo controller could also be improved. The servo controllers can operate in an open loop mode and allow for an external controller to be used. Code has already been developed for an external motor controller.

A fault detection and tolerance algorithm could be tested on the simulator. If one of the reaction wheels where to fail, the only change to the controller would be in redefining the rotation matrices G and H.

References

[1] Blenden, R., *Regenerative Power-Optimal Reaction Wheel Attitude Control*, Master's thesis, University of Colorado, 2010.

[2] Swartwout, M., "The First One Hundred CubeSats: A Statistical Look," *Journal of Small Satellites*, 2013, pp. 213–233.

[3] Sommer, J., *Development of an On-board Data Handling System and a Star Camera for a Nano Satellite Simulation Table*, Master's thesis, Lulea University of Technology, 2013.

[4] Samuels, M. A., *The Design and Testing of a Thee-Degree-of-Freedom Small Satellite Simulator using a Linear Controller with Feedback Linearization and Trajectory Generation*, Master's thesis, Utah State University, 2014.

[5] Sidi, M. J., *Spacecraft Dynamics and Control*, Cambridge Univ. Press, 1997.

[6] Wertz, J. R., *Spacecraft Attitude Determination and Control*, Reidel, 1986.

[7] Crowell, C. W., *Development and Analysis of a Small Satellite Attitude Determination and Control System Testbed*, Master's thesis, Massachusetts Institute of Technology, 2011.

[8] Prado, J., "Three axis air-bearing based platform for small satellite attitude determination and control simulation," *Journal of Applied Research and Technology*, 2005.

[9] Janson, S., "Attitude Control on the Pico Satellite Solar Cell Testbed-2," *26th Annual AIAA/USU Conference on Small Satellites*, 2012.

[10] Kang, W., "Co-ordinated attitude control of multi-satellite systems," *International Journal of Robust and Nonlinear Control*, 2002, pp. 185–205.

[11] Boynton, R., "Using A Spherical Air Bearing to Simulate Weightlessness," *55th Annual Conference of the Society of Allied Weight Engineers*, 1996.

[12] Dorf, R. C., *Modern Control Systems*, Pearson Prentice Hall, Upper Saddle River, NJ, 2005.

[13] Fullmer, R. R., "Dynamic Ground Testing of the Skipper Attitude Control System," *34th AIAA Aerospace Sciences Meeting and Exhibit*, 1996.

[14] Lord Microstrain, "3DM-GX3-25," `http://files.microstrain.com/3DM-GX3-25-Attitude-Heading-Reference-System-Data-Sheet.pdf`, December 2014.

[15] Maxon Motor, "EC 45 flat," `http://www.maxonmotor.com/medias/sys_master/8813854621726/14-233-EN-Jun.pdf`, December 2014.

Appendices

Appendix A

IMU Data Sheet

Source: [14]

3DM-GX3® -25

Miniature Attitude Heading Reference System

The **3DM-GX3®** -25 is a high-performance, miniature Attitude Heading Reference System (AHRS), utilizing MEMS sensor technology. It combines a triaxial accelerometer, triaxial gyro, triaxial magnetometer, temperature sensors, and an on-board processor running a sophisticated sensor fusion algorithm to provide static and dynamic orientation, and inertial measurements.

Features & Benefits

Best in Class

- precise attitude estimations
- high-speed sample rate & flexible data outputs
- high performance under vibration and high *g*

Easiest to Use

- smallest, lightest industrial AHRS available
- simple integration supported by SDK and comprehensive API

Cost Effective

- reduced cost and rapid time to market for customer's applications
- aggressive volume discount schedule

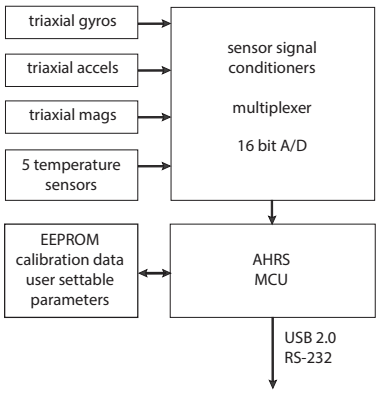
Applications

Accurate guidance, orientation and positioning under dynamic conditions such as:

- Inertial Aiding of GPS
- Unmanned Vehicle Navigation
- Platform Stabilization, Artificial Horizon
- Antenna and Camera Pointing
- Health and Usage Monitoring of Vehicles
- Reconnaissance, Surveillance, and Target Acquisition
- Robotic Control
- Personnel Tracking

System Overview

The **3DM-GX3®** -25 offers a range of fully calibrated inertial measurements including acceleration, angular rate, magnetic field, deltaTheta and deltaVelocity vectors. It can also output computed orientation estimates including Euler angles (pitch, roll, and heading (yaw)), rotation matrix and quaternion. All quantities are fully temperature compensated and are mathematically aligned to an orthogonal coordinate system. The angular rate quantities are further corrected for g-sensitivity and scale factor non-linearity to third order. The 3DM-GX3® -25 architecture has been carefully designed to substantially eliminate common sources of error such as hysteresis induced by temperature changes and sensitivity to supply voltage variations. Gyro drift is eliminated in AHRS mode by referencing magnetic North and Earth's gravity and compensating for gyro bias. On-board coning and sculling compensation allows for use of lower data output rates while maintaining performance of a fast internal sampling rate. For those users, integrators or OEMs who develop their own orientation and navigation applications, the 3DM-GX3® -25 is shipped with a complete Data Communications Protocol guide that provides access to the powerful LORD MicroStrain® Inertial Packet Protocol (MIP). Applications of your own design can readily be developed in any coding language and on any computing platform including microprocessors. The 3DM-GX3® -25 is initially sold as a starter kit consisting of an AHRS+GPS module, RS-232 or USB communication and power cable, software CD, user manual and quick start guide.



best of sensors expo 2009
Silver Level Winner

LORD MicroStrain®
SENSING SYSTEMS

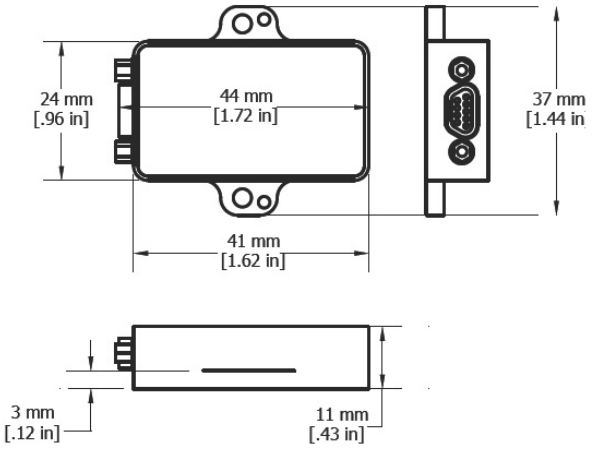
Specifications

AHRS Specifications

Attitude and Heading	
Attitude heading range	360° about all 3 axes
Accelerometer range	±5*g* standard
Gyroscope range	±300°/sec standard
Static accuracy	±0.5° pitch, roll, heading typical for static test conditions
Dynamic accuracy	±2.0° pitch, roll, heading for dynamic (cyclic) test conditions and for arbitrary angles
Long term drift	eliminated by complimentary filter architecture
Repeatability	0.2°
Resolution	<0.1°
Data output rate	up to 1000 Hz
Filtering	sensors sampled at 30 kHz, digitally filtered (user adjustable) and scaled into physical units; coning and sculling integrals computed at 1 kHz
Output modes	acceleration, angular rate, and magnetic field deltaTheta, deltaVelocity, Euler angles, quaternion, rotation matrix
General	
A/D resolution	16 bits SAR oversampled to 17 bits
Interface options	USB 2.0 or RS232
Baud rate	115,200 bps to 921,600 bps
Power supply voltage	+3.2 to +16 volts DC
Power consumption	80 mA @ 5 volts with USB
Connector	micro-DB9
Operating temperature	-40° C to +70° C
Dimensions	44 mm x 24 mm x 11 mm - excluding mounting tabs, width across tabs 37 mm
Weight	18 grams
ROHS	compliant
Shock limit	500 *g*
Software utility	CD in starter kit (XP/Vista/Win7/Win8 compatible)
Software development kit (SDK)	complete data communications protocol and sample code

Senror Specifications

	Accels	Gyros	Mags
Measurement range	±5 *g*	±300°/sec	±2.5 Gauss
Non-linearity	±0.1 % fs	±0.03 % fs	±0.4 % fs
In-run bias stability	±0.04 m*g*	18°/hr	—
Initial bias error	±0.002 *g*	±0.25°/sec	±0.003 Gauss
Scale factor stability	±0.05 %	±0.05 %	±0.1 %
Noise density	80 μ*g*/√Hz	0.03°/sec/√Hz	100 μGauss/√Hz
Alignment error	±0.05°	±0.05°	±0.05°
User adjustable bandwidth	225 Hz max	440 Hz max	230 Hz max
Sampling rate	30 kHz	30 kHz	7.5 kHz max
Options			
Accelerometer range	±1.7 *g*, ±16 *g*, ±50 *g*		
Gyroscope range	±50°/sec, ±600°/sec, ±1200°/sec		



Copyright © 2014 LORD Corporation
Strain Wizard®, DEMOD-DC®, DVRT®, DVRT-Link™, WSDA®, HS-Link®, TC-Link®, G-Link®, V-Link®, SG-Link®, ENV-Link™, Watt-Link™, Shock-Link™, LXRS®, Node Commander®, SensorCloud™, Live Connect™, MathEngine®, EH-Link®, 3DM®, FAS-A®, 3DM-GX3®, 3DM-DH®, 3DM-DH3™, MicroStrain®, and Little Sensors, Big Ideas.® are trademarks of LORD Corporation.
Specifications are subject to change without notice.

8400-0033 rev. 003

LORD Corporation
MicroStrain® Sensing Systems
459 Hurricane Lane,
Suite 102
Williston, VT 05495 USA
www.microstrain.com
ph: 800-449-3878
fax: 802-863-4093
sales@microstrain.com
Patent Pending

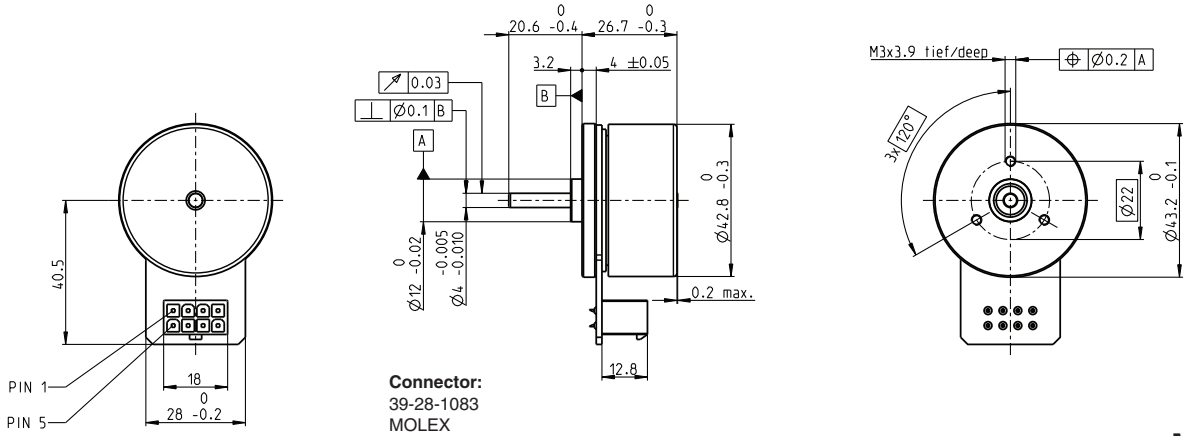
Appendix B

Motor Data Sheet

Source: [15]

EC 45 flat ∅45 mm, brushless, 70 Watt

NEW



Connector:
39-28-1083
MOLEX

M 1:2

Stock program
Standard program
Special program (on request)

Order Number

with Hall sensors		397172	402685	402686	402687
Motor Data (provisional)					
Values at nominal voltage					
1 Nominal voltage	V	24	30	36	48
2 No load speed	rpm	6110	6230	6330	3440
3 No load current	mA	234	194	166	48.1
4 Nominal speed	rpm	4840	4970	5060	2540
5 Nominal torque (max. continuous torque)	mNm	130	114	110	134
6 Nominal current (max. continuous current)	A	3.26	2.4	1.96	0.935
7 Stall torque	mNm	820	783	773	738
8 Starting current	A	39.5	25.8	20.7	6.97
9 Max. efficiency	%	85	84	83	84
Characteristics					
10 Terminal resistance phase to phase	Ω	0.608	1.16	1.74	6.89
11 Terminal inductance phase to phase	mH	0.463	0.691	0.966	5.85
12 Torque constant	mNm / A	36.9	45.1	53.3	131
13 Speed constant	rpm / V	259	212	179	72.7
14 Speed / torque gradient	rpm / mNm	4.26	5.44	5.85	3.82
15 Mechanical time constant	ms	8.07	10.3	11.1	7.24
16 Rotor inertia	gcm²	181	181	181	181

Specifications

Thermal data

17	Thermal resistance housing-ambient	3.39 K / W
18	Thermal resistance winding-housing	4.0 K / W
19	Thermal time constant winding	28.8 s
20	Thermal time constant motor	157 s
21	Ambient temperature	-40 ... +100°C
22	Max. permissible winding temperature	+125°C

Mechanical data (preloaded ball bearings)

23	Max. permissible speed	10000 rpm
24	Axial play at axial load < 4.0 N	0 mm
	> 4.0 N	0.14 mm
25	Radial play	preloaded
26	Max. axial load (dynamic)	3.8 N
27	Max. force for press fits (static)	50 N
	(static, shaft supported)	1000 N
28	Max. radial loading, 7.5 mm from flange	21 N

Other specifications

29	Number of pole pairs	8
30	Number of phases	3
31	Weight of motor	141 g

Values listed in the table are nominal.

Connection

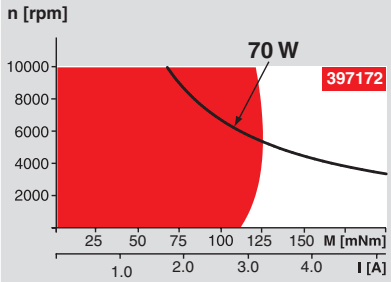
Pin 1	Hall sensor 1*
Pin 2	Hall sensor 2*
Pin 3	4.5 ... 18 VDC
Pin 4	Motor winding 3
Pin 5	Hall sensor 3*
Pin 6	GND
Pin 7	Motor winding 1
Pin 8	Motor winding 2

*Internal pull-up (7 … 13 kΩ) on pin 3
Wiring diagram for Hall sensors see p. 29

Cable

Connection cable Universal, L = 500 mm	**339380**
Connection cable to EPOS, L = 500 mm	**354045**

Operating Range



Comments

Continuous operation
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

Short term operation
The motor may be briefly overloaded (recurring).

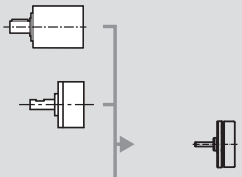
Assigned power rating

maxon Modular System

Overview on page 16 - 21

Planetary Gearhead
∅42 mm
3 - 15 Nm
Page 237

Spur Gearhead
∅45 mm
0.5 - 2.0 Nm
Page 239



Recommended Electronics:

DECS 50/5	Page 289
DEC 24/3	290
DEC 50/5	291
DEC Module 50/5	291
DECV 50/5	297
EPOS2 Module 36/2	304
EPOS2 24/5	305
EPOS2 P 24/5	308
Notes	**20**

Option
With Cable and Connector
(Ambient temperature -20 ... +100°C)