Development and Test of a Kinematical Method for Estimating Rotation Using the Acceleration of a Rigid Body

Douglas A. Carr

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation
DEVELOPMENT AND TEST OF A KINEMATICAL METHOD FOR ESTIMATING ROTATION USING THE ACCELERATION OF A RIGID BODY

By Douglas A. Carr

A Thesis submitted in Partial Fulfillment of the Requirement for the

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

Approved By:

Dr. Wayne W. Walter
Department of Mechanical Engineering

Dr. Agamemnon L. Crassidis
Department of Mechanical Engineering

Dr. Ferat Sahin
Department of Electrical Engineering

Dr. Edward C. Hensel
Department Head of Mechanical Engineering

DEPARTMENT OF MECHANICAL ENGINEERING ROCHESTER INSTITUTE OF TECHNOLOGY

SEPTEMBER, 2006
Permission Granted:

Development and Test of a Kinematical Method for Estimating Rotation Using the Acceleration of a Rigid Body

I, Douglas A. Carr, hereby grant permission to the Wallace Library of the Rochester Institute of Technology to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Date: ___________ Signature of Author: Douglas A. Carr
Abstract

In this work, the feasibility of a novel method for combining inertial information gathered from accelerometers to calculate an estimate for rotational information independent of rate gyro measurements is investigated. The concept uses two, distinctly placed, parallel accelerometers on a rigid body to estimate rotational acceleration by comparing the difference in sensed acceleration caused by rotation. Since the estimation is independent of available rate gyro data, optimization of drift and scale factors can be used to calibrate the accelerometers in a real time fashion. Experimental trials were conducted on a small mobile robotic platform, due to the eventual objective of using the kinematical hypothesis described above to develop an improved, compact Inertial Navigation System, ideal for autonomous robotic navigation in hazardous environments where Global Positioning signal are not available.

Preliminary trials were conducted in a two-dimensional environment, and accelerometer data was used to estimate heading and generate positional maps of the paths traveled. The estimated rotation data was also used to roughly determine the scale and drift factors of the accelerometer inputs, which were independently measured for verification. Important conclusions about the effects of instrument placement geometry were reached, and the developed system was scrutinized for recommended improvements for the next generation of research.
Acknowledgements

I would like to thank my fiancéé Alaina for her patience and support throughout this process, without which it might never have come to fruition. I would like to thank my advisors: Dr. Walter for his guidance and infinite perseverance and Dr. Crassidis for his inspiration in the classroom and his willingness to challenge me as a student. Both have made a significant impact on my life. I am much indebted to Rob Kraynik of the ME shop, for the countless answered questions and hours upon hours of help, his expertise was invaluable. My three surrogate moms in the ME office, Diane, Connie, and Sheila deserve much of the credit for helping keep things in perspective, and for looking out for me, probably in more ways than I know.

I am eternally grateful to my parents in more ways than I could ever list.

And finally I would like to acknowledge all the friends and teachers that enriched my life in Rochester. From shared late nights to moral support and even physical shelter, I’m truly fortunate to know so many wonderful people. Erin Long and Eric Linden especially will both have my eternal gratitude and, of course, a place on my couch should they ever desire it.
Table of Contents

Abstract................................................................................................................................. i

Acknowledgements ........................................................................................................... ii

Table of Contents ............................................................................................................... iii

List of Figures ....................................................................................................................... vi

List of Tables ......................................................................................................................... ix

Nomenclature ....................................................................................................................... x

1 Introduction ..................................................................................................................... 1

2 Project History ............................................................................................................... 8

3 Theory .......................................................................................................................... 14

3.1 MIMNS ...................................................................................................................... 14

3.2 Two Dimensional Case ............................................................................................. 24

3.3 MIMNS Via Velocities ............................................................................................. 25

3.4 r – Feedback ............................................................................................................... 26

3.5 Non-Orthogonal Instrument Axes ............................................................................ 27

3.6 Rigid body assumption ............................................................................................. 32
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.7</td>
<td>Encoder Model</td>
<td>34</td>
</tr>
<tr>
<td>4</td>
<td>Current Hardware Development</td>
<td>39</td>
</tr>
<tr>
<td>4.1</td>
<td>Wheels &amp; Encoders</td>
<td>39</td>
</tr>
<tr>
<td>4.2</td>
<td>Motion Control</td>
<td>44</td>
</tr>
<tr>
<td>4.3</td>
<td>Environmental Sensing</td>
<td>47</td>
</tr>
<tr>
<td>4.4</td>
<td>MIMNS Data Collection</td>
<td>51</td>
</tr>
<tr>
<td>4.5</td>
<td>Onboard Software</td>
<td>56</td>
</tr>
<tr>
<td>5</td>
<td>Offline Software Development</td>
<td>61</td>
</tr>
<tr>
<td>5.1</td>
<td>MIMNS Kalman Filter</td>
<td>61</td>
</tr>
<tr>
<td>5.2</td>
<td>Simulation</td>
<td>68</td>
</tr>
<tr>
<td>5.3</td>
<td>Updated Kalman Filter</td>
<td>72</td>
</tr>
<tr>
<td>5.4</td>
<td>Map Making</td>
<td>78</td>
</tr>
<tr>
<td>6</td>
<td>Data Collection and Results</td>
<td>82</td>
</tr>
<tr>
<td>6.1</td>
<td>Troubleshooting</td>
<td>82</td>
</tr>
<tr>
<td>6.2</td>
<td>Testing</td>
<td>90</td>
</tr>
<tr>
<td>6.3</td>
<td>Further Improvements</td>
<td>95</td>
</tr>
<tr>
<td>6.4</td>
<td>Final Implementation</td>
<td>98</td>
</tr>
</tbody>
</table>
7 Conclusions and Recommendations ......................................................... 119

8 Reference List ......................................................................................... 122

9 Appendices .............................................................................................. 126

9.1 Appendix A - Original Kalman Filter Simulation, MATLAB .m-file ............ 126

9.2 Appendix B – Modified Kalman Filter .m-file ................................................. 130

9.3 Appendix C – MIMNS .m-file ................................................................. 134

9.4 Simulink Diagram for r-feedback methods .............................................. 137

9.5 Appendix D – ADXL202 and ADXRS150 Datasheets ............................... 138
List of Figures

Figure 1 - Gimbaled Inertial Platform [11] ................................................................. 6
Figure 2 - Original robot ............................................................................................... 10
Figure 3 – Spring 2003 Robot Map [19] ................................................................... 12
Figure 4 – Fall 2004 Robot Map [21] ....................................................................... 13
Figure 5 - MIMNS cube ............................................................................................... 16
Figure 6 - MIMNS definitions ..................................................................................... 17
Figure 7 - Location vectors ......................................................................................... 19
Figure 8 - Non-Orthogonal Axis ................................................................................. 28
Figure 9 - Non-orthogonal IMU axis system .............................................................. 29
Figure 10 - MIMNS cube mechanical drawing [19] .................................................. 32
Figure 11 - Vehicle Model ............................................................................................ 34
Figure 12 - Incremental Turn ....................................................................................... 35
Figure 13 - Turning Wheel ........................................................................................... 36
Figure 14 - PC6 Timing Diagram (PC6-83-1) ............................................................ 40
Figure 15 - Hardware Schematic ................................................................................ 42
Figure 16 - PWM signal .............................................................................................. 45
Figure 17 - View of the Current Robot ....................................................................... 46
Figure 18 – Layout Schematic of Current Robot ....................................................... 46
Figure 19 - Correction angle diagram ......................................................................... 48
Figure 20 - Correction Angle Derivation ................................................................... 48
Figure 21 - PCB Layout ............................................................................................... 53
Figure 22 - Rate Gyro Test

Figure 23 - PCB Schematic

Figure 24 - Motion Control flow chart

Figure 25 - Data collection flow chart

Figure 26 - Noise simulation

Figure 27 - First Kalman simulation

Figure 28 - Updated Kalman Filter Simulations

Figure 29 - Theoretical Standard Deviation in Kalman Estimate

Figure 30 - Simulated Global Map

Figure 31 - Encoder Map: Straight Line Test

Figure 32 - Encoder trial map

Figure 33 - First Run

Figure 34 - Motor Test 1

Figure 35 - Optical De-coupler

Figure 36 - Walking Experiment

Figure 37 - Accelerometer Functional Block Diagram

Figure 38 - Accel Data with Lowered Bandwidth

Figure 39 - Improved Raw MIMNS Data

Figure 40 - Relocated Accelerometer

Figure 41 - Accels on outer walls

Figure 42 - Updated PCB Schematic

Figure 43 - Results From New PCBs

Figure 44 - Measurement Details
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td>Kalman Estimate, Zero Process Noise</td>
<td>100</td>
</tr>
<tr>
<td>46</td>
<td>Kalman Estimate with Varying Process Noise</td>
<td>100</td>
</tr>
<tr>
<td>47</td>
<td>Final Kalman Estimate</td>
<td>101</td>
</tr>
<tr>
<td>48</td>
<td>Global INS Map of RIT Hallway</td>
<td>102</td>
</tr>
<tr>
<td>49</td>
<td>Encoder Map of RIT Hallway</td>
<td>104</td>
</tr>
<tr>
<td>50</td>
<td>Raw Acceleration and Velocity Data</td>
<td>105</td>
</tr>
<tr>
<td>51</td>
<td>Accel Anomaly Test Map</td>
<td>107</td>
</tr>
<tr>
<td>52</td>
<td>Anomaly Test, Raw Data</td>
<td>108</td>
</tr>
<tr>
<td>53</td>
<td>MIMNS pt. 1</td>
<td>109</td>
</tr>
<tr>
<td>54</td>
<td>MIMNS pt. 2</td>
<td>112</td>
</tr>
<tr>
<td>55</td>
<td>Accelerator Optimization Block Diagram</td>
<td>116</td>
</tr>
<tr>
<td>56</td>
<td>MIMNS Optimized Parameters</td>
<td>118</td>
</tr>
<tr>
<td>57</td>
<td>MIMNS Simulink Block Diagram</td>
<td>137</td>
</tr>
</tbody>
</table>
List of Tables

Table 1 – Exported data example ............................................................................. 60
## Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Acceleration</td>
<td>ft/s²</td>
</tr>
<tr>
<td>(a_{rel})</td>
<td>Relative acceleration</td>
<td>ft/s²</td>
</tr>
<tr>
<td>(a_{x,i}, a_{y,i}, a_{z,i})</td>
<td>Measured accelerations transformed to an orthogonal axis system.</td>
<td>ft/s²</td>
</tr>
<tr>
<td>(\alpha)</td>
<td>Rotational acceleration vector</td>
<td>ft/s²</td>
</tr>
<tr>
<td>(a_{\alpha,j})</td>
<td>Total acceleration of instrument</td>
<td>ft/s²</td>
</tr>
<tr>
<td>A</td>
<td>Jacobian of non-linear system model</td>
<td>n/a</td>
</tr>
<tr>
<td>A_k</td>
<td>Discrete system dynamics matrix</td>
<td>n/a</td>
</tr>
<tr>
<td>(A_{\alpha,j})</td>
<td>Pure linear acceleration of instruments</td>
<td>ft/s²</td>
</tr>
<tr>
<td>A/D</td>
<td>Analog to Digital</td>
<td>n/a</td>
</tr>
<tr>
<td>ADC</td>
<td>Analog to Digital Converter</td>
<td>n/a</td>
</tr>
<tr>
<td>ASME</td>
<td>American Society of Mechanical Engineers</td>
<td>n/a</td>
</tr>
<tr>
<td>BGA</td>
<td>Ball Grid Array</td>
<td>n/a</td>
</tr>
<tr>
<td>B</td>
<td>Gyro bias state</td>
<td>n/a</td>
</tr>
<tr>
<td>(\beta)</td>
<td>Correction angle</td>
<td>degrees</td>
</tr>
<tr>
<td>(C_t)</td>
<td>Center of turning</td>
<td>n/a</td>
</tr>
<tr>
<td>CW</td>
<td>Clockwise</td>
<td>n/a</td>
</tr>
<tr>
<td>CCW</td>
<td>Counter-Clockwise</td>
<td>n/a</td>
</tr>
<tr>
<td>(d\beta)</td>
<td>Incremental turning angle (Encoder Model)</td>
<td>degrees</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Units</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>(d_{\theta i})</td>
<td>Incremental traveled angle of wheel (Encoder Model)</td>
<td>degrees</td>
</tr>
<tr>
<td>(d\psi)</td>
<td>Incremental heading change (Encoder Model)</td>
<td>degrees</td>
</tr>
<tr>
<td>(d s_r)</td>
<td>Right incremental arclength (Encoder Model)</td>
<td>ft</td>
</tr>
<tr>
<td>(d s_l)</td>
<td>Left incremental arclength (Encoder Model)</td>
<td>ft</td>
</tr>
<tr>
<td>DCM</td>
<td>Duty Cycle Modulated</td>
<td>n/a</td>
</tr>
<tr>
<td>DIP</td>
<td>Dual Inline Package</td>
<td>n/a</td>
</tr>
<tr>
<td>(E)</td>
<td>Expected value function</td>
<td>n/a</td>
</tr>
<tr>
<td>EKF</td>
<td>Extended Kalman Filter</td>
<td>n/a</td>
</tr>
<tr>
<td>FEA</td>
<td>Finite Element Analysis</td>
<td>n/a</td>
</tr>
<tr>
<td>(\Gamma_{\text{accel},i}), (\Gamma_{\text{gyro},i})</td>
<td>Misalignment translation matrices</td>
<td>n/a</td>
</tr>
<tr>
<td>(g)</td>
<td>Gravity (Approx 32.2 ft/s²)</td>
<td>ft/s²</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
<td>n/a</td>
</tr>
<tr>
<td>(H)</td>
<td>Measurement matrix</td>
<td>n/a</td>
</tr>
<tr>
<td>I/O</td>
<td>Input/Output</td>
<td>n/a</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial Measurement Unit</td>
<td>n/a</td>
</tr>
<tr>
<td>INS</td>
<td>Inertial Navigation System</td>
<td>n/a</td>
</tr>
<tr>
<td>(J_{\rho})</td>
<td>Optimization cost function</td>
<td>n/a</td>
</tr>
<tr>
<td>(k)</td>
<td>Time step counter</td>
<td>n/a</td>
</tr>
<tr>
<td>(K_k)</td>
<td>Kalman gain</td>
<td>n/a</td>
</tr>
<tr>
<td>(l)</td>
<td>Distance from CT to left wheel (Encoder Model)</td>
<td>ft</td>
</tr>
<tr>
<td>(L)</td>
<td>Hypotenuse of diagonal sensor readings</td>
<td>ft</td>
</tr>
<tr>
<td>MIMNS</td>
<td>Micro-Inertial Measurement/Navigation System</td>
<td>n/a</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Unit</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>MEMS</td>
<td>Micro Electro-Mechanical System</td>
<td>n/a</td>
</tr>
<tr>
<td>O</td>
<td>Moving origin</td>
<td>n/a</td>
</tr>
<tr>
<td>$\Phi_k$</td>
<td>Fundamental system matrix</td>
<td>n/a</td>
</tr>
<tr>
<td>$\Phi_s$</td>
<td>System noise constant</td>
<td>n/a</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Pitch (Euler angle)</td>
<td>degrees</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Roll (Euler angle)</td>
<td>degrees</td>
</tr>
<tr>
<td>$\psi$</td>
<td>Heading (Yaw) (Euler angle)</td>
<td>degrees</td>
</tr>
<tr>
<td>$p$</td>
<td>Rotation rate about X axis (pitch rate)</td>
<td>deg/s</td>
</tr>
<tr>
<td>$\dot{p}$</td>
<td>Rotational acceleration about X</td>
<td>deg/s²</td>
</tr>
<tr>
<td>$\ddot{p}$</td>
<td>MIMNS estimation of rotation rate about X axis</td>
<td>deg/s</td>
</tr>
<tr>
<td>$P_k$</td>
<td>Covariance matrix</td>
<td>n/a</td>
</tr>
<tr>
<td>PCB</td>
<td>Printed Circuit Board</td>
<td>n/a</td>
</tr>
<tr>
<td>$q$</td>
<td>Rotation rate about Y axis (roll rate)</td>
<td>deg/s</td>
</tr>
<tr>
<td>$\dot{q}$</td>
<td>Rotational acceleration about Y</td>
<td>deg/s²</td>
</tr>
<tr>
<td>$\ddot{q}$</td>
<td>MIMNS estimation of rotation rate about Y axis</td>
<td>deg/s</td>
</tr>
<tr>
<td>$Q_c$</td>
<td>Continuous process noise matrix</td>
<td>n/a</td>
</tr>
<tr>
<td>$Q_k$</td>
<td>Discreet process noise matrix</td>
<td>n/a</td>
</tr>
<tr>
<td>$r$</td>
<td>Rotation rate about Z axis (yaw rate)</td>
<td>deg/s</td>
</tr>
<tr>
<td>r</td>
<td>Distance from $C_t$ to right wheel (ENCODER MODEL ONLY)</td>
<td>ft</td>
</tr>
<tr>
<td>$\dot{r}$</td>
<td>Rotational acceleration about Z</td>
<td>deg/s²</td>
</tr>
</tbody>
</table>

xii
\( \bar{r} \)
MIMNS estimation of rotation rate about \( Z \) axis deg/s

\( r_l \)
Left wheel turning radius (Encoder Model) ft

\( r_r \)
Right wheel turning radius (Encoder Model) ft

\( r_{rel} \)
Relative distance from common to instrument axis ft

\( R \)
Measurement noise matrix n/a

**RAM**
Random Access Memory n/a

**RIT**
Rochester Institute of Technology n/a

**RLG**
Ring Laser Gyro n/a

\( \rho \)
Instantaneous turning radius ft

\( s \)
Position estimate ft

\( s_k \)
Linear distance traveled by \( C_t \) at time \( k \) ft

**SLAM**
Simultaneous Localization And Mapping n/a

\( v_{a,i} \)
Velocity of instrument ft/s

\( v_{rel} \)
Relative velocity ft/s

\( V_v \)
Vehicle velocity ft/s

\( \omega \)
Rotational rate deg/s

\( \omega \)
Rotation rate vector deg/s

\( w_l \)
Left wheel radius ft

\( w_r \)
Right wheel radius ft

**W**
Optimization output weighting matrix n/a

**WGN**
White Gaussian Noise n/a
Kalman estimate at \( k \)

System model prediction of current state

Reference frame axes of "moving origin" \( O \)

Instrument axis system, 1\(^{st}\) accel in \( \alpha \) direction

Instrument axis system, 2\(^{nd}\) accel in \( \alpha \) direction

Distance components from inertial frame to moving origin \( O \)

Distance components from inertial frame to instrument frame

Accelerometer misalignment angles

Gyro misalignment angles

Measurement vector


Introduction

In the rapidly advancing fields of research collectively known as robotics, many applications exist where accurate positional knowledge in a global frame are required. Traditionally, the potential uses for robotic technology have been limited to applications where high repetition of manual tasks or inhospitable environments rendered human labor excessively costly or impractical. Often, limited computing power dictated the sensitivity and information content of available sensor inputs and rendered complex analysis unacceptably time consuming. Early industrial robotic applications included “pick and place” tasks where machines such as the PUMA and SCARA robotic arms perform controlled, repetitive movements based on simple sensor inputs, [1,2] as well as more complex methodical tasks such automobile manufacturing and painting, where three-dimensional directions or patterns are “taught” into the machines memory by a human expert [3]. In these cases, any significant variation in the robot’s task require a human operator to make alterations to the machine’s program code and to re-teach system variables, such as movement waypoints. However, the systems still found widespread use due to the advantages gained in the ability to engineer desirable properties into a robotic system, such as high strength or fatigue resistance, operability in extreme temperatures or resistance to corrosive or toxic environments.

Early academic research of autonomous robotic behavior began in strictly controlled environments with goals of narrow scope. Tasks that required any significant decision making were relegated to machines teleo-operated by human controllers, or else faced prohibitively large processing times. For example, the Stanford Cart, developed at
Stanford University in the 1970’s was capable of following a painted white line at a reasonable pace. Later Cart developments, including a prototype vision analysis system, allowed the robot to navigate a 30-meter obstacle course autonomously; however, the time required to transverse the course exceeded approximately five hours [4].

Other early robotic applications included NASA’s interplanetary missions of the 1970’s and 80’s, such as the Mars Viking landers. Developed by Jet Propulsion Laboratories in Pasadena, CA, the pair of Viking explorers “had a 6000 word memory for command instructions” that were received through an antenna from earth-based control centers and were carried out until the next batch of communications was received [5]. The landers were successfully inserted onto the surface of Mars and operated through a wide range of scientific missions there; however, the missions were necessarily limited in scope. “The delay for radio signals between Mars and Earth vary between 6 to 41 minutes while the long distance imposes a low communication bandwidth. This precludes the use of teleoperation for controlling the vehicle,” meaning that with the technology available at the time, neither remote human control nor endogenous robotic control were viable options for movement across the surface of Mars [6].

The last two decades have seen rapid improvements in computing capabilities and development of sensing technologies. As such, one of the major thrusts of robotics research since the late 1980’s has been utilizing the available technology to effect autonomous navigation. Much effort has been spent on what is coined the “Simultaneous Localization and Mapping (SLAM) problem. “[SLAM] is, in many respects, a ‘Holy
Grail’ of the autonomous vehicle research community. The ability to place an autonomous vehicle at an unknown location in an unknown environment and then have it build a map, using only relative observations of the environment, and then to use this map simultaneously to navigate would indeed make such a robot ‘autonomous’ [7].”

The most common modern approach to addressing the SLAM problem was first proposed in 1989 by Smith et al.. This work introduced the concept of a stochastic map, focusing around spatial variables, which are vectors containing location and orientation data of an object relative to a global reference frame. The robot system calculates a landmark’s spatial variable, as well as the uncertainty in the calculations by “estimating the first two moments of its probability distribution – the mean, and the covariance.” [8] This approach lends itself well to the use of Kalman Filter or Extended Kalman Filter as a means of iteratively estimating the robot’s states, and the Kalman technique has become one of the most commonly used estimation algorithms in robotic navigation applications. A key point made by the stochastic map approach was that the certainty with which a robotic system determines the locations of each of the landmarks used to navigate is correlated between various landmarks due to the common uncertainty in the location of the robot with respect to the global coordinate system. In other words, since the robot system does not know its precise location in the global coordinate system, even if perfect measurements to the relative locations of landmarks are available, uncertainty in the global location of those landmarks is still a given. Previous approaches had disregarded these cross-correlations, with the result that data analysis by the robot indicated more information than what was truly available.
Regardless of the approach used, accurate knowledge of the robot’s position information in a global frame is a critical element for navigation. This is true not only in the development of ground based robotic technologies, such as those addressed in this paper, but of any field where navigation information is required, such as aeronautics and astronautics. Indeed one of the core considerations for any SLAM navigation approach must be the certainty with which a machine can estimate its own position. There are many methods of obtaining this information, such as utilizing the GPS satellite constellation, or analysis of image processing, odometry or pedometry, or inertial measurement. Each of these methods has strengths and weaknesses, and for a localization method to be robust, usually several must be employed in tandem. For example, in circumstances where the use of GPS is not dependable, such as when direct view of the satellites is not available (e.g. operating indoors, underground, in a forested area), or, more dramatically, on the surface of a celestial body, a navigation routine may rely primarily on inertial navigation system information for localization knowledge.

Inertial navigation is a category of widely used methods of estimating location, “which implement Newton’s laws of motion… [utilizing] inertial instruments (i.e. accelerometers and gyroscopes) to sense vehicle acceleration in a known direction and integrate this acceleration to determine velocity and position [9].” The drawback is that typically the overall error in the estimate grows unbounded with time, since even small errors in acceleration measurement are twice integrated over time to find position, resulting in an estimate that rapidly diverges from true position. For instance, one relatively high
accuracy platform used in nautical navigation has an error budget of approximately 0.5-1.0 nmi/hr [10]. Section 2.1 includes further exploration of the functionality of the individual inertial sensors (accelerometers and gyros.)

Inertial Navigation Systems (INS) can be divided into two broad categories based on their mode of functioning: gimbaled INS, where the sensors are suspended within a gimbal ring device, and "strapdown" INS, where the devices are firmly fixed (i.e. strapped down) to the body of moving vehicle [11]. A gimbaled INS involves a gimbaled platform onto which rate gyros and accelerometers are mounted for detection in an orthogonal axis pattern. The outputs from the gyros are used to control compensating torque motors on each of the gimbaled axes, so that the instrument platform remains oriented parallel to the inertial (i.e. unmoving or observer) reference frame, regardless of the rotation of the supporting body. Accelerometer outputs are then integrated once to yield velocity estimates, and again to yield position estimates.
Figure 1 - Gimbaled Inertial Platform [11]

This arrangement can be very effective and reliable; however, due to its inherent size, mechanical complexity, and associated cost, gimbaled systems are not appropriate candidates for an INS mounted on a small mobile robot, such as the one discussed herein [11]. Strapdown INS concepts emerged in the early 1970’s as a means of eliminating the need for gimbals altogether. Rather than using gyros to control the inertial stability of the sensor platform, the estimates simply maintain a knowledge of the orientation of accelerometers as they rotate in space. Accelerations can then be transformed to an inertial reference frame, in essence a “mathematical gimbal set” to replace the mechanical one [11]. The advent of Micro Electro-Mechanical System (MEMS) accelerometers and gyros has further increased the potential for development of small, lightweight, and inexpensive strapdown INS systems.
To date, however, MEMS inertial technology is relatively crude in its sensitivity (compare error accumulation of 70°/hr in a modern solid state MEMS gyro versus .015°/hr in a much older, more traditional Ring Laser Gyro (RLG) [10,12],) and so numerous methods of compensation have been explored for improving the accuracy of the estimates. These include the incorporation of accurate vehicle system models, typically through a Kalman filter ([13-15]), and mathematical compensation for known sources of error, such as gravity related effects, through a variety of means [9,16,17].

The hypothesis explored in this work uses two MEMS accelerometers to characterize the rotation of the body to which they are attached, with the intention of using the gathered information to improve the estimate of spatial variables. Accordingly, the work has been given the title Micro Inertial Measurement and Navigation System, or MIMNS. The content was also presented at the ASME International Mechanical Engineering Congress and Exposition at Orlando in November 2005 [31].
1 Project History

The developments described in this thesis are intended to represent just one link in a chain of research that has the eventual goal of developing robust and inexpensive miniaturized robotic systems capable of autonomous navigation and mapmaking, with or without the aid of a GPS receiver. Several teams and individual students at RIT have made contributions to this research, and it is important to note the development history in order to fully understand the conditions that shape the current stage of development.

The project began in 2003 with the building of MIMNS cube by Copeland et al. at RIT "in preparation for research into a theory that the signals from an inertial navigation system’s accelerometers can be used to estimate error in rate gyro signals." [19] The project team successfully designed and built the thermoplastic cube superstructure as well as the first generation of integrated circuits containing accelerometers and gyros to be installed on the cube’s faces. The sensors used were the ADXRS150 MEMS gyro and ADXL210 dual-axis MEMS accelerometer, both from Analog Devices. The XRS150 outputs a voltage proportional to the rate of rotation of the device, with a range of ±150°/s, while each of the two perpendicular accelerometers in the XL210 output a voltage proportional to the acceleration in their respective directions, up to ±10 g (where g is approximately 32.2 ft/s, the average acceleration due to the earth’s gravity). At the conclusion of this project, the team was able to gather some signal output from the instruments using a LabView workstation from National Instruments, but significant testing was left to future researchers. The team’s final product is pictured in “Figure 5 - MIMNS cube.”
The following year a new team was commissioned to create an autonomous robotic unit capable of navigating and mapping the hallways in RIT’s academic buildings utilizing the MIMNS cube developed during the previous year. [20] The result of their efforts was a small roving robot with a footprint of approximately 18-in by 15-in, standing approximately four inches tall, pictured in “Figure 2 - Original robot.” The chassis was machined from .075-in sheet aluminum in a semi-protective style, with the underside, left and right sides of the inner compartment protected by the chassis while the front, back and top are open to the environment. The mechanical drive train consisted of four wheels, with the right and left side pairs driven independently by two 12-volt DC motors. This setup enabled “skid” style steering, in which the wheels on one side are turned in the opposite direction as the wheels on the opposing side, causing the robot to turn in place with little translation, but also causing significant chattering as the wheels skid sideways through a turn. In preparation for this thesis, this chattering was observed both visibly and audibly during trial runs, and became an immediate item of concern.
The original navigation algorithm consisted of a basic obstacle avoidance routine, which utilized three sensors mounted on the front end of the chassis. In the center, an ultrasonic rangefinder was mounted on a pulse-width modulated servo motor that rotated to take readings at ±45° angles, as well as straight ahead. As a precaution, two infrared proximity detectors were also mounted on the robot’s front end at the outer edges, detecting any obstacles directly in front of the robot that might have escaped detection by the ultrasonic rangefinder. If the infrared sensors were tripped, the robot would come to an immediate stop and initiate an avoidance routine.

An MSP430 micro-controller from Texas Instruments utilized the center rangefinder readings to determine if the path ahead of the robot had been blocked, and if so whether to turn left or right, or simply backup (in the case where forward, left and right paths were all blocked.) The MSP430 would then activate the motors to move forward, backward, turn left or turn right, using a pair of H-bridge type relay circuits. Each
motion routine was timed, such that the robot would turn approximately \( \pm 90^\circ \) per rotation command or move forward or backward an approximate distance per translation command.

In the original design, the MSP430 micro-controller was used to collect data from the MIMNS unit as well as to control the motion control algorithm described above; however, due to timing issues, a second processor was added separate the tasks (a DIOS micro-controller.) The new controller was selected because of it’s built in A/D capabilities which reduced hardware requirements thus simplifying the data acquisition process. The DIOS received acceleration and rotation rate inputs from the sensors on the MIMNS cube and performed a numerical integration to approximate distances traveled and orientation angles, which were then stored in memory. These stored data points were then extracted once the robot had been retrieved and processed into a map of the robot’s location. An example of a map generated by the robot of the third floor of the James E. Gleason building in the Kate Gleason College of Engineering is shown in Figure 3; however, neither the original data sets nor the collection, processing, or post-processing routines used to generate this map have survived, hence its authenticity cannot be directly verified.
During the summer and fall of 2004, design modifications were made by an RIT graduate student, including replacing the existing micro-controllers with BS2 Basic Stamp micro-controllers, and replacing the two forward facing proximity detectors with outward facing (to the right and left at ± 90° to the robot centerline) ultrasonic rangefinders [21]. The most significant change was the re-design of the MIMNS printed circuit boards, replacing the ADXL210 accelerometers with the ADXL202 model. The XL202 has a detection range of ± 2 g, yielding better sensitivity to the small accelerations sensed by the robot, and having a duty cycle modulated (DCM) output specifically designed for easy interaction with low cost micro-controllers eliminating the need for extra ADC’s. The DCM outputs a constant period square wave with the duty cycle (as defined in Section 3.2) proportional to the acceleration so that a micro-controller with a timer can decode the output by timing the pulses and calculating the duty cycle.
Combining range data recorded by the side-looking ultrasonic sensors with location and orientation data, a map of the third floor of the James E. Gleason building was generated, shown in Figure 4. The map is an improvement to the previously developed map shown in Figure 3, with increased detail outlining the hallway network; however, as with the last map, the original data and post processing routines are not available for verification of the authenticity of this work.

Figure 4 – Fall 2004 Robot Map [21]
2 Theory

This chapter outlines the theoretical foundation of the mathematical models and supporting assumptions that were used in the research and experimentation presented in later chapters. The equations that form a basis for the Micro Inertial Measurement and Navigation System (MIMNS) are presented in Section 2.1, and assumptions and simplifications made during the development are subsequently shown in Sections 2.2 - 2.6. Finally, the vehicle model used to interpret data from the wheel encoders is derived in Section 2.7.

2.1 MIMNS

The position and orientation of a rigid body in three-dimensional space are typically described relative to some inertial reference frame using Cartesian coordinates and Euler angles. Inertial sensing is one method of determining defining coordinates using sensors such as accelerometers, which output a signal proportional to acceleration (denoted ‘\(a\)’), and rate gyros, whose output is proportional to rotational rate (‘\(\omega\)’). These signals can then be integrated over time yielding estimates of position (‘\(s\)’) and orientation (‘\(\psi\)’) according to

\[
s(t) = \int \int a(t) \, dt^2
\]

\[
\psi(t) = \int \omega(t) \, dt
\]

The accuracy of the relative position and orientation information obtained by this approach is determined by sensitivity, drift, and noise properties of the instruments. For example, inertial instruments typically are effected by a drifting signal bias, meaning that
the output (usually a voltage) is consistently offset from its nominal predicted (true) value, yielding measurements which, when integrated over long periods of time, cause an unbounded error in a position estimate. The biases are generally a property of the individual instrument as a result of tolerances in manufacturing, and can vary over time due to many factors such as changes in temperature. For rate gyro output to be meaningful in a situation such as aerial navigation, the drift in the instrument’s readings must be significantly smaller than the rotation rate of the earth (360°/24-hr or .00417°/sec) or else that rotation will escape detection, creating possible flaws in an avionic navigation algorithm [18]. In the case of the accelerometers, since double integration is required in the position estimate, position estimates are extremely sensitive to even very small signal biases. Historically, these considerations lead to the selection of extremely accurate and sensitive (and therefore extremely expensive) instruments in applications where inertial sensing is a requirement.

The approach being investigated at RIT [19-21] utilizes information gathered from accelerometers to estimate the rotation of a body measured independently from a gyroscope. The additional information realized from these estimates could then be potentially used to improve the accuracy of an Inertial Measurement Unit (IMU) platform, for instance by comparing the MIMNS estimate to the rate gyro output for the same period, the error between the two can be used to identify potential sources of accelerometer error such as drift and bias (see Section 5.4.3.) Figure 5 displays a prototype Micro Inertial Measurement and Navigation System (MIMNS) cube device developed at RIT for the purpose of testing the proposed hypothesis.
The device is conceptualized in Figure 6, which illustrates the cube with accelerometers placed for measurement in the x-direction, in opposite corners of opposing faces of the cube. For clarity of illustration, the y-direction and z-direction accelerometers are omitted. Rate gyros measuring rotation about the X, Y and Z axis for comparison were integrated into the device shown in Figure 5; however, since their relative placement is arbitrary and not necessarily confined to the faces of the cube, the gyros are not shown in Figure 6.
A coordinate system, denoted \{X,Y,Z\} has been aligned with 3 of the faces of the cube, with the origin shown by \(O\). The origin, \(O\), is a "moving origin," since it translates relative to an inertial reference frame, and its orientation is used to define relative motion between a rotating body (in this case the cube) and that same inertial reference frame. The axis system is typically aligned with a vehicle body axis, center of gravity, or center of rotation depending on the application. Rotations rates about the \(X\), \(Y\), and \(Z\)-axes are denoted \(p\), \(q\) and \(r\) respectively, and the time derivatives of these rotations rates are
similarly located and sensed, though not illustrated in Figure 6. The Euler angles \( \Phi, \theta, \text{ and } \Psi \) represent the pitch, roll and heading angles of the device (for a detailed description of Euler angles, refer to [22]). The rotation rate and acceleration components form vectors \( \omega \) and \( \alpha \), i.e.

\[
\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad \alpha = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}
\] (2.1.3)

Each instrument has an associated axis system, typically located at the center of the physical instrument itself, and the direction in which the instrument senses is aligned with the corresponding axis. For example in Figure 6, the accelerometer \( a_{x,1} \), which senses in the X direction, is located at the origin of the \( \{x_{x,1}, y_{x,1}, z_{x,1}\} \) instrument axis system, and is aligned (by definition) to sense acceleration along the \( x_{x,1} \) axis. For clarity only the “x” accelerometers are shown in the figure, but in similar fashion a “y” accelerometer \( a_{y,2} \) would be located at the \( \{x_{y,2}, y_{y,2}, z_{y,2}\} \) instrument axis system, and aligned to sense along the \( y_{y,2} \) axis, etc.

The relative distance between the common and instrument axis systems with respect to an arbitrary inertial reference frame can be represented in vector form as:

\[
r_{rel} = \begin{bmatrix} x_{a,i} - \bar{x} \\ y_{a,i} - \bar{y} \\ z_{a,i} - \bar{z} \end{bmatrix}
\] (2.1.4)

As denoted in Figure 7, \( \bar{x}, \bar{y}, \bar{z} \) represent distance components from an arbitrary inertial coordinate frame (e.g. a fixed point on the earth’s surface with the proper assumptions) to
the moving origin $O$, and $\vec{x}_{a,i}, \vec{y}_{a,i}, \vec{z}_{a,i}$ represent distance components from the inertial origin to the instrument frame, where \( \alpha \) is accelerometer direction \((x, y, z)\) and \(i\) is accelerometer number \((1, 2)\).

The derivation of the equations for estimating rotation rate begins with the classical analytical expression of relative acceleration. [23]

$$\mathbf{a} = \mathbf{a}_0 + \mathbf{a} \times \mathbf{r}_{rel} + \mathbf{\omega} \times (\mathbf{\omega} \times \mathbf{r}_{rel}) + 2\mathbf{\omega} \times \mathbf{v}_{rel} + \mathbf{a}_{rel} \tag{2.1.5}$$

where $\mathbf{a}_0$ is the translational acceleration of the moving origin $O$, while $\mathbf{v}_{rel}$ and $\mathbf{a}_{rel}$ are the velocity and acceleration components between points on the body (i.e. relative to the moving origin axis system \(\{X,Y,Z\}\)) due to flexion. Making a rigid body assumption (reference Section 2.6 “Rigid body assumption”), $\mathbf{v}_{rel}$ and $\mathbf{a}_{rel}$ terms tend towards zero, and after substituting labels from Figure 6, Equation (2.1.5) reduces to
\[ \mathbf{a}_{a,i} = \mathbf{A}_{a,i} + \mathbf{a} \times \mathbf{r}_{rel} + \mathbf{\omega} \times (\mathbf{\omega} \times \mathbf{r}_{rel}) \] (2.1.6)

\( \mathbf{A}_{a,i} \) is the purely linear acceleration component of the body, i.e. the acceleration of the point \( \mathbf{O} \), which is not influenced by rotation rates, and \( \mathbf{a}_{a,i} \) is the total acceleration as observed from the inertial reference frame and recorded by the instrument. Substituting Equation (2.1.4), carrying out the cross products, and rearranging yields

\[ \mathbf{a}_{a,i} = \mathbf{A}_{a,i} + \left[ \begin{array}{c}
-(q^2 + r^2) \cdot (\bar{x} - \bar{x}_{a,i}) - (\dot{r} - pq)(\bar{y} - \bar{y}_{a,i}) + (\dot{q} + rp)(\bar{z} - \bar{z}_{a,i}) \\
(\dot{r} + pq) \cdot (\bar{x} - \bar{x}_{a,i}) - (p^2 + r^2)(\bar{y} - \bar{y}_{a,i}) - (\dot{p} - qr)(\bar{z} - \bar{z}_{a,i}) \\
-(\dot{q} - pr) \cdot (\bar{x} - \bar{x}_{a,i}) + (\dot{p} + qr)(\bar{y} - \bar{y}_{a,i}) - (p^2 + q^2)(\bar{z} - \bar{z}_{a,i})
\end{array} \right] \] (2.1.7)

Recall in the case of the \( \mathbf{A}_{x,i} \) accelerometers, the instruments are aligned with the \( x_{x,i} \) axes, and similarly, the \( \mathbf{A}_{y,i} \) accelerometers are aligned with the \( y_{y,i} \) axes and the \( \mathbf{A}_{z,i} \) accelerometers are aligned with the \( z_{z,i} \) axes. Each of the three "i" components from the "i" instrument axis systems are combined forming an orthogonal basis for an IMU (Inertial Measurement Unit) axis system \( \{x_{x,i}, y_{y,i}, z_{z,i}\} \), and can then be transformed to the common reference frame \( \{X, Y, Z\} \), given knowledge of the Euler angles. The orthogonality of the IMU axis system is an important assumption for the MIMNS cube device, since the axis systems in Figure 6 are assumed parallel; however, in practice this is rarely realized to due imperfections in the construction of the device. A method for handling with the non-orthogonal IMU axis systems is presented in Section 2.5 "Non-Orthogonal Instrument Axes."

Combining the appropriate components from the first "i" accelerometer triad, a complete description of the transformed accelerations is derived:
\[ a_{x,1} = A_{x,1} - (q^2 + r^2) \cdot (\vec{x} - \vec{x}_{x,1}) - (\dot{r} - pq)(\vec{y} - \vec{y}_{x,1}) + (\dot{q} + rp)(\vec{z} - \vec{z}_{x,1}) \]
\[ a_{y,1} = A_{y,1} + (\dot{r} + pq) \cdot (\vec{x} - \vec{x}_{y,1}) - (p^2 + r^2)(\vec{y} - \vec{y}_{y,1}) - (\dot{p} - qr)(\vec{z} - \vec{z}_{y,1}) \]
\[ a_{z,1} = A_{z,1} - (\dot{q} - pr) \cdot (\vec{x} - \vec{x}_{z,1}) + (\dot{p} + qr)(\vec{y} - \vec{y}_{z,1}) - (p^2 + q^2)(\vec{z} - \vec{z}_{z,1}) \]

Similarly for the second accelerometer triad transformed to the common system:

\[ a_{x,2} = A_{x,2} - (q^2 + r^2) \cdot (\vec{x} - \vec{x}_{x,2}) - (\dot{r} - pq)(\vec{y} - \vec{y}_{x,2}) + (\dot{q} + rp)(\vec{z} - \vec{z}_{x,2}) \]
\[ a_{y,2} = A_{y,2} + (\dot{r} + pq) \cdot (\vec{x} - \vec{x}_{y,2}) - (p^2 + r^2)(\vec{y} - \vec{y}_{y,2}) - (\dot{p} - qr)(\vec{z} - \vec{z}_{y,2}) \]
\[ a_{z,2} = A_{z,2} - (\dot{q} - pr) \cdot (\vec{x} - \vec{x}_{z,2}) + (\dot{p} + qr)(\vec{y} - \vec{y}_{z,2}) - (p^2 + q^2)(\vec{z} - \vec{z}_{z,2}) \]

Subtracting Equation (2.1.9) from Equation (2.1.8) yields

\[ (a_{x,1} - a_{x,2}) = (A_{x,1} - A_{x,2}) - (q^2 + r^2) \cdot (\vec{x}_{x,2} - \vec{x}_{x,1}) - (\dot{r} - pq)(\vec{y}_{x,2} - \vec{y}_{x,1}) + (\dot{q} + rp)(\vec{z}_{x,2} - \vec{z}_{x,1}) \]
\[ (a_{y,1} - a_{y,2}) = (A_{y,1} - A_{y,2}) + (\dot{r} + pq) \cdot (\vec{x}_{y,2} - \vec{x}_{y,1}) - (p^2 + r^2)(\vec{y}_{y,2} - \vec{y}_{y,1}) - (\dot{p} - qr)(\vec{z}_{y,2} - \vec{z}_{y,1}) \]
\[ (a_{z,1} - a_{z,2}) = (A_{z,1} - A_{z,2}) - (\dot{q} - pr) \cdot (\vec{x}_{z,2} - \vec{x}_{z,1}) + (\dot{p} + qr)(\vec{y}_{z,2} - \vec{y}_{z,1}) - (p^2 + q^2)(\vec{z}_{z,2} - \vec{z}_{z,1}) \]

(2.1.10)

The acceleration components \( A_{x,1} \) and \( A_{x,2} \) represent the \( x \)-direction linear accelerations of the center of rotation or center of gravity of the rigid body to which both accelerometers are attached, and as such they are equal so that

\[ (A_{x,1} - A_{x,2}) = 0 \]
\[ (A_{y,1} - A_{y,2}) = 0 \]
\[ (A_{z,1} - A_{z,2}) = 0 \]

(2.1.11)

Substituting Equation (2.1.11) into Equation (2.1.10) and solving the \( x \) component for \( \dot{r} \),
the \( y \) component for \( \dot{p} \) and the \( z \) component for \( \dot{q} \) results in

21
\[
\dot{r} = pq + \frac{-(a_{x,1} - a_{x,2}) + (\dot{q} + rp)(\bar{z}_{x,2} - \bar{z}_{x,1}) - (q^2 + r^2)(\bar{x}_{x,2} - \bar{x}_{x,1})}{\bar{y}_{x,2} - \bar{y}_{x,1}}
\]
\[
\dot{p} = qr + \frac{-(a_{y,1} - a_{y,2}) + (\dot{r} + pq)(\bar{z}_{y,2} - \bar{z}_{y,1}) - (p^2 + r^2)(\bar{y}_{y,2} - \bar{y}_{y,1})}{\bar{z}_{y,2} - \bar{z}_{y,1}}
\]
\[
\dot{q} = pr + \frac{-(a_{z,1} - a_{z,2}) + (\dot{p} + qr)(\bar{z}_{z,2} - \bar{z}_{z,1}) - (p^2 + q^2)(\bar{z}_{z,2} - \bar{z}_{z,1})}{\bar{x}_{z,2} - \bar{x}_{z,1}}
\]

(2.1.12)

As a simplification, the following notation is substituted into Equation (2.1.12).

\[
(\bar{x}_{x,2} - \bar{x}_{x,1}) = \bar{x}, \quad (\bar{x}_{y,2} - \bar{x}_{y,1}) = \bar{y}, \quad (\bar{x}_{z,2} - \bar{x}_{z,1}) = \bar{z}
\]
\[
(\bar{y}_{x,2} - \bar{y}_{x,1}) = \bar{y}, \quad (\bar{y}_{y,2} - \bar{y}_{y,1}) = \bar{y}, \quad (\bar{y}_{z,2} - \bar{y}_{z,1}) = \bar{y}
\]
\[
(\bar{z}_{x,2} - \bar{z}_{x,1}) = \bar{z}, \quad (\bar{z}_{y,2} - \bar{z}_{y,1}) = \bar{z}, \quad (\bar{z}_{z,2} - \bar{z}_{z,1}) = \bar{z}
\]

(2.1.13)

Resulting in

\[
\dot{r} = pq - \frac{(a_{x,1} - a_{x,2}) + (\dot{q} + rp)\bar{z}}{\bar{y}} - \frac{(q^2 + r^2)\bar{x}}{\bar{y}}
\]
\[
\dot{p} = qr - \frac{(a_{y,1} - a_{y,2}) + (\dot{r} + pq)\bar{x}}{\bar{z}} - \frac{(p^2 + r^2)\bar{y}}{\bar{z}}
\]
\[
\dot{q} = pr - \frac{(a_{z,1} - a_{z,2}) + (\dot{p} + qr)\bar{z}}{\bar{x}} - \frac{(p^2 + q^2)\bar{z}}{\bar{x}}
\]

(2.1.14)

Finally, for an initialization step, by substituting the expression for \( \dot{p} \) into the equation for \( \dot{q} \) and then substituting the new expression for \( \dot{q} \) into the equation for \( \dot{r} \) and solving the resulting equation for \( \dot{r} \) we arrive at
\[ \dot{r} = \frac{-(a_{x,1} - a_{x,2}) \overline{x} \overline{z} - (q^2 + r^2) \overline{x} \overline{x} \overline{z} + 2 pr \overline{z} \overline{z} \overline{y}}{M} \]
\[ + \frac{-(a_{y,1} - a_{y,2}) \overline{y} \overline{z} - (p^2 + r^2) \overline{y} \overline{y} \overline{z} + 2 qr \overline{y} \overline{z} \overline{y}}{M} \]
\[ + \frac{-(a_{z,1} - a_{z,2}) \overline{z} \overline{z} - (p^2 + q^2) \overline{z} \overline{z} \overline{z} + pq (\overline{x} \overline{y} \overline{z} + \overline{x} \overline{y} \overline{z})}{M} \]  
(2.1.15)

\[ \dot{p} = qr - \frac{(a_{y,1} - a_{y,2}) \overline{z} \overline{y}}{\overline{z} \overline{y}} + \frac{(\dot{r} + pq) \overline{x} \overline{y}}{\overline{x} \overline{y}} - \frac{(p^2 + r^2) (\overline{y} \overline{y})}{\overline{y} \overline{y}} \]

\[ \dot{q} = pr - \frac{(a_{z,1} - a_{z,2}) \overline{x} \overline{z}}{\overline{x} \overline{z}} + \frac{(\dot{p} + qr) \overline{y} \overline{z}}{\overline{y} \overline{z}} - \frac{(p^2 + q^2) \overline{z}}{\overline{z}} \]

\[ M = x_2 y_1 y_3 - x_1 y_2 z_3 \]

Note that once a numerical value of \( \dot{r} \) has been calculated the value is substituted into the subsequent equations, completing calculation for \( \dot{p} \) and \( \dot{q} \). Integrating the angular acceleration terms from Equation (2.1.15) yields an estimate for the angular rates derived from accelerometer information:

\[ \ddot{r} = \int \dot{r} \, dt \]

\[ \ddot{p} = \int \dot{p} \, dt \]

\[ \ddot{q} = \int \dot{q} \, dt \]  
(2.1.16)

where \( \ddot{p} \), \( \ddot{q} \), and \( \ddot{r} \) are the estimated rates and are now be fed back into Equation (2.1.15). This is an important result, since the derived angular rate is not dependant on the angular position of the device, which is very difficult to estimate accurately. The derived rate information from Equation (2.1.15) can now be combined with gyro instrument outputs to estimate accelerometer bias and scale factors, therefore determining a more accurate estimation of the true position (see Section 5.4.3.)
If the true rotational rate is known accurately then accelerometer biases can be estimated by minimizing the difference between the angular rate estimates and gyro outputs using an optimization routine such as the "simplex search routine" in MATLAB's "fminsearch" function, or any comparable routine [24]. Typically an optimization routine varies the input parameters (in this case the input parameters are the accelerometer biases or other error factors) to minimize a cost function such as the one suggested in Equation (2.1.17):

\[ J_\omega = \int [\omega - \tilde{\omega}]^T W [\omega - \tilde{\omega}] dt \] (2.1.17)

where \( \omega \) is the measurement vector of rate gyro outputs and \( \tilde{\omega} \) is the vector of rate estimates, varying with changing accelerometer biases; \( J_\omega \) is minimized when \( [\omega - \tilde{\omega}] \) is minimized, i.e. when the chosen biases cause the calculated estimate to closely match the measured rate gyro signal. When a minimum for \( J_\omega \) is found, the input parameters are considered to be at their optimal values. The output weighting matrix, \( W \), allows the designer, if desired, to place more emphasis on finding input parameters matching a particular component of the vector.

### 2.2 Two Dimensional Case

The platform used to test the hypothesis functions primarily is a flat-space environment such as a building hallway. Due to this restriction and to simplify the testing of the hypothesis, a two dimensional case is assumed. An extension to the three dimensional case can be considered once the two dimensional case is verified. However, only a two dimensional case is considered in this work. For a two dimensional environment, altitude \( z \), pitch angle \( \theta \), and roll angle \( \phi \) as shown in Figure 6 are assumed to be zero. Since these values are identically zero, their derivatives must also be zero so that
\[ p = 0 \quad , \quad \dot{p} = 0 \]
\[ q = 0 \quad , \quad \dot{q} = 0 \]
\[ \dot{z} = 0 \quad , \quad \ddot{z} = 0 \]  \hspace{1cm} (2.2.1)

Substituting Equation (2.2.1) into the expression for \( \dot{\mathbf{r}} \) in Equation (2.1.14), yields

\[ \dot{\mathbf{r}} = \frac{(a_{s,2} - a_{s,1})}{\overline{y}_x} \frac{r^2 \cdot \overline{x}}{\overline{y}_x} \]  \hspace{1cm} (2.2.2)

The simplified equation shown above is used in the post processing routines for the analysis described in Section 4.

### 2.3 MIMNS Via Velocities

As a hypothesis, the velocity of each accelerometer can be estimated by integrating the acceleration signals over time according to

\[ v_{a,i}(t) = \int_0^t a_{a,i}(\tau) d\tau \]  \hspace{1cm} (2.3.1)

A similar process described in Section 2.1 for estimating \( p, q, \) and \( r \) is developed using the well known velocity equation, derived in [23]:

\[ \mathbf{v} = \mathbf{v}_0 + \omega \times \mathbf{r}_{rel} + \mathbf{v}_{rel} \]  \hspace{1cm} (2.3.2)

where \( \mathbf{v} \) is the true velocity who’s components are defined by Equation (2.3.1), \( \mathbf{v}_0 \) is the velocity of the center of rotation, \( \omega \) and \( \mathbf{r}_{rel} \) are defined in Section 2.1, with \( \mathbf{v}_{rel} \) set equal to zero due to the rigid body assumption. Carrying out the cross product and rearranging:
\[
\begin{align*}
  v_{x,i} &= v_{o,x} + q(z-x_i) - r(y-y_i) \\
  v_{y,i} &= v_{o,y} + r(x-x_i) - p(z-z_i) \\
  v_{z,i} &= v_{o,z} + p(y-y_i) - q(x-x_i)
\end{align*}
\] (2.3.3)

Subtracting signal two from signal one, and substituting the notation from (2.1.13) yields

\[
\begin{align*}
  (v_{x,i} - v_{x,2}) &= q \cdot z_x - r \cdot y_x \\
  (v_{y,i} - v_{y,2}) &= r \cdot z_y - p \cdot z_y \\
  (v_{z,i} - v_{z,2}) &= p \cdot y_z - q \cdot x_z
\end{align*}
\] (2.3.4)

and finally, solving for the rotational elements yields

\[
\begin{align*}
  r &= \frac{(v_{x,i} - v_{x,2}) + q \cdot z_x}{y_x} \\
  p &= \frac{(v_{y,i} - v_{y,2}) + r \cdot z_y}{z_y} \\
  q &= \frac{(v_{z,i} - v_{z,2}) + p \cdot y_z}{x_z}
\end{align*}
\] (2.3.5)

Implementation of this method is further discussed in Section 5.4.

2.4 \textit{r – Feedback}

The second term in Equation (2.2.2) contains an \( \text{"} r^2 \text{"} \), resulting from centripetal acceleration caused by rotation. Several potential methods for feeding back this rotation information are investigated in the results Section (5.4) and are listed here.

**Direct Feedback** – As shown in Equation (2.1.16), the calculated rotational acceleration vector \( \alpha \), defined in Equation (2.1.3), can be integrated over time, giving an estimate for rotational velocity \( \omega \), also defined in Equation (2.1.3). At time step \( k \), the rotation estimate from time step \( k-1 \) can be fed into the MIMNS equation.
Estimate from Velocities – Applying a two dimensional assumption to Equation (2.3.5) simplifies the expression for $r$ to $r = (v_{x,1} - v_{x,2})/y_x$. After performing the necessary numerical integration using the current time acceleration measurement, the instantaneous estimate for $r$ can be calculated and substituted into the expression for rotational acceleration.

Gyro – If a signal from a rate gyro is available, it can be substituted directly.

2.5 Non-Orthogonal Instrument Axes

As mentioned in Section 2.1, an assumption is made for the MIMNS cube device that the $x$, $y$ and $z$ instrument axis combine to form an orthogonal instrument axis system. In practice however, this may not always be the case. For example, manufacturing of the cube device and or mounting of the instruments results in imperfections where the instrument axes become non-orthogonal. Furthermore, installation of the device on a system such as an aircraft results in misalignment errors since the device will not be perfectly aligned with the body axes of the aircraft. Figure 8 illustrates a case where such a misalignment has occurred on the $a_{x,1}$ accelerometer.
Though not illustrated, the $a_y$ and $a_z$ accelerometer may also be misaligned, each with its own unique orientation. Furthermore, the three gyros used to measure $p$, $q$ and $r$ may be misaligned with the $\{X,Y,Z\}$ axis system. Note that the instrument axis system itself is orthogonal by definition, being located and aligned with the physical instrument; however, when combining the $x_{x,1}$, $y_{y,1}$, and $z_{z,1}$ axes (the sensing axis from each instrument system), each with its own unique misalignment angles, the resultant IMU axis system is non-orthogonal, as is illustrated in Figure 9.
Figure 9 - Non-orthogonal IMU axis system

The set of sensed accelerations from the misaligned reference frame must be transformed to an equivalent set \((a_{tx,i}, a_{ty,i}, a_{tz,i})\), parallel to the common reference frame, for accurate measurement. The misalignment in each instrument axis system can be described in terms of misalignment angles, measured relative to the common reference system \((X, Y, Z)\). For example the misaligned \(a_{x,i}\) accelerometer in Figure 8 has misalignment angles \((\psi_{x,i}, \theta_{x,i}, \phi_{x,i})\), and a misaligned gyro for measuring rotational rate \(p\) has misalignment angles \((\psi_p, \theta_p, \phi_p)\). Using the misalignment angle transformations, the misaligned, sensed acceleration can be described in terms of components of the true, orthogonal set \((a_{tx,i}, a_{ty,i}, a_{tz,i})\), through the associated misalignment angles \((\psi_{x,i}, \theta_{x,i}, \phi_{x,i})\). The transformation is (see reference [22]):
\[ a_{x,i} = a_{x,i} \cos(\theta_{x,i}) \cos(\psi_{x,i}) + a_{y,i} \cos(\theta_{y,i}) \sin(\psi_{x,i}) - a_{z,i} \sin(\theta_{x,i}) \]  \hfill (2.5.1)

Similarly, the y and z acceleration vectors can be transformed using:

\[ a_{y,i} = a_{x,i} \left[ \cos(\psi_{y,i}) \sin(\theta_{y,i}) \sin(\phi_{y,i}) - \sin(\psi_{y,i}) \cos(\phi_{y,i}) \right] \]
\[ + a_{y,i} \left[ \sin(\phi_{y,i}) \sin(\psi_{y,i}) \sin(\theta_{y,i}) + \cos(\psi_{y,i}) \cos(\phi_{y,i}) \right] \]
\[ + a_{z,i} \sin(\phi_{y,i}) \cos(\theta_{y,i}) \]

\[ a_{z,i} = a_{x,i} \left[ \cos(\psi_{z,i}) \cos(\phi_{z,i}) \sin(\theta_{z,i}) + \sin(\psi_{z,i}) \sin(\phi_{z,i}) \right] \]
\[ + a_{y,i} \left[ \sin(\phi_{z,i}) \cos(\psi_{z,i}) \sin(\theta_{z,i}) - \cos(\psi_{z,i}) \sin(\phi_{z,i}) \right] \]
\[ + a_{z,i} \cos(\theta_{z,i}) \cos(\phi_{z,i}) \]  \hfill (2.5.2)

A similar approach is used to transform the misaligned rate gyros:

\[ p = p_{r} \cos(\theta_{r}) \cos(\psi_{r}) + q_{r} \cos(\theta_{r}) \sin(\psi_{r}) - r_{r} \sin(\theta_{r}) \]
\[ q = p_{r} \left[ \cos(\psi_{q}) \sin(\theta_{q}) \sin(\phi_{q}) - \sin(\psi_{q}) \cos(\phi_{q}) \right] \]
\[ + q_{r} \left[ \sin(\phi_{q}) \sin(\psi_{q}) \sin(\theta_{q}) + \cos(\psi_{q}) \cos(\phi_{q}) \right] \]
\[ + r_{r} \sin(\phi_{q}) \cos(\theta_{q}) \]  \hfill (2.5.3)
\[ r = p_{r} \left[ \cos(\psi_{r}) \cos(\phi_{r}) \sin(\theta_{r}) + \sin(\psi_{r}) \sin(\phi_{r}) \right] \]
\[ + q_{r} \left[ \sin(\psi_{r}) \cos(\phi_{r}) \sin(\theta_{r}) - \cos(\psi_{r}) \sin(\phi_{r}) \right] \]
\[ + r_{r} \cos(\phi_{r}) \cos(\theta_{r}) \]

In Equation (2.5.3) \( p, q, \) and \( r \) are the measured gyro outputs while \( p_{r}, q_{r}, \) and \( r_{r} \) are the rotations transformed to the \( \{X,Y,Z\} \) reference frame.

Mounting the unit on a test stand, and subjecting it to known input accelerations and rotations \( a_{x,0}, a_{y,0}, a_{z,0}, p_{0}, q_{0}, \) and \( r_{0} \) while instrument data is recorded, an optimization routine can be performed, varying the misalignment angles for each individual instrument.
to find the best fit between the measured and the true data. Once these angles are known, they can be substituted into equations (2.5.2) and (2.5.3), and an inversion is performed to give the mathematical model for translating misaligned instruments into an aligned, orthogonal axis system, i.e.,

\[
\begin{align*}
\vec{a}_{\alpha,i} & = \left[ \Gamma_{\text{accel},i} \right]^{-1} \vec{a}_{x,i} \\
\vec{a}_{\beta,i} & = \left[ \Gamma_{\text{accel},i} \right]^{-1} \vec{a}_{y,i} \\
\vec{a}_{\gamma,i} & = \left[ \Gamma_{\text{accel},i} \right]^{-1} \vec{a}_{z,i}
\end{align*}
\]  
\hspace{1cm} (2.5.4)

\[
\begin{align*}
\vec{p} & = \left[ \Gamma_{\text{gyro}} \right]^{-1} \vec{p}_i \\
\vec{q} & = \left[ \Gamma_{\text{gyro}} \right]^{-1} \vec{q}_i \\
\vec{r} & = \left[ \Gamma_{\text{gyro}} \right]^{-1} \vec{r}_i
\end{align*}
\]  
\hspace{1cm} (2.5.5)

where the \(\Gamma\)'s are the translation matrices containing the inverted misalignment angle components. Simulations performed in MATLAB using Equations (2.5.1), (2.5.2) and (2.5.3) indicate that if all three of an instrument's misalignment Euler angles vary by five degrees, the overall change effected by the misalignment will be less than 1% of the total input. For a worst case scenario where all misalignment Euler angles vary by 10 degrees, the maximum change in the output was approximately 3.4% of the total input. Given that the MIMNS cube was designed with tight tolerances to minimize misalignments (±0.005-in), and since mounting on the autonomous robotic platform is relatively easy to control, the assumption that the individual misalignment angles are kept within a five degree tolerance is valid, thereby making the effect of misalignment smaller than 1% of total input.
2.6 Rigid body assumption

During design and development of the MIMNS cube, several iterations of analysis were conducted before the current design was selected. The material for the cube is a thermoplastic polyoxymethylene (POM), having several desirable properties for this application including high mechanical strength (average 64.9 MPA tensile yield, 110 MPA compressive yield) and hardness, and good vibration damping properties. The design goal for the project was for the structure to be able to withstand a 10g acceleration with minimal internal stresses and deflections. Figure 10 shows an engineering drawing of the implemented design.

Figure 10 - MIMNS cube mechanical drawing [19]

FEA analysis was performed on this structure using IDEAS by constraining the base in a vertical orientation and applying a 10-g downward body force for analyzing the cantilever beam properties of the structure, i.e., the weakest mode of deflection.
The simulations predicted that the maximum Von Mises stress resulting from the applied 10-g body force was approximately $3.63 \times 10^{-2}$ psi, while the maximum deflection was $1.99 \times 10^{-7}$ in. [19]. It was estimated using Equation (2.6.1) that the minimum deflection sensed by the accelerometers was approximately $1.54 \times 10^{-4}$ in., a full three orders of magnitude larger than maximum deflection induced by a 10-g acceleration.

$$s = \frac{1}{2} a_o t^2$$  \hspace{1cm} (2.6.1)

The unit has since been refitted with accelerometers that were designed for a smaller range of detection ($\pm 2$-g vs. $\pm 10$-g), for an increased resolution. Reapplying the method used in [19] to the new accelerometers, the deflection was determined using equation (2.6.1), where “s” is the deflection, and assuming 0.8-mg resolution at 50-Hz (according to the instrument data sheet [25]) in the current design, $a_o = (32.2 \times 8 \times 10^{-3}) \text{ ft/s}^2$ and $t = (1/50) s$, giving a minimum detectable deflection of $5.152 \times 10^{-6}$ in. This number, while considerably smaller than minimum detectable deflection of the original instruments, is still an order of magnitude greater than the largest deflection experienced during a 10-g acceleration.

It is also noteworthy that the cube was originally designed to be placed in an airplane where accelerations as high as 10-g might possibly be encountered. In its current application, the $\pm 2$-g accelerometers are more than adequate to capture the expected accelerations caused by the motion of the robot, and so, since the FEA was done under a
±10-g assumption, the calculations above describe not only a worst case scenario, but one with a considerable safety factor.

2.7 Encoder Model

As is described in Section 3.1, two encoders and accompanying equipment were installed to provide a comparison of the IMU data to "dead reckoned" position estimates, i.e., those based on the rotation of the wheels. The most significant shortcomings of dead-reckoned estimates are attributed to "non-systematic" errors; i.e. those errors not predicted in advance, for example wheel slippage [26]. A vehicle model must first be developed to be able to create a map using the encoder data. The following model is based on work conducted at Virginia Tech [26] for modeling a robotic drive system similar in concept to the one presented here.

Figure 11 - Vehicle Model

Figure 11 shows a top view representation of the robot with dimensions illustrated. The center of turning (‘C_t’) is a fixed point on the chassis centerline at a point directly between the centers of the left and right wheels. The distances to the left and right
wheels (‘l’ and ‘r’ respectively) and the left and right wheel radii (\( w_l \) and \( w_r \) respectively) are kept distinct since these values may not be exactly the same due to variations in chassis and wheel geometry.

**Figure 12 - Incremental Turn**

Figure 12 illustrates the path of the vehicle moving through an incremental turn (‘\( d\beta \)’), ‘\( d\psi \)’ represents the incremental heading change experienced during the turn, while ‘\( \rho \)’ represents the instantaneous turning radius. The left and right wheels move through arcs ‘\( ds_l \)’ and ‘\( ds_r \)’, respectively, located at the left and right wheel path radii ‘\( r_l \)’ and ‘\( r_r \)’ from the instantaneous center of rotation. ‘\( V_v \)’ is the instantaneous velocity of the vehicle’s center of turning, \( C_t \).
Figure 13 - Turning Wheel

By inspection of the geometry of Figure 12, the angle ‘dΨ’ is equal to the angle ‘dθ’.

We can also note that the arc length ‘ds_i’ is equal to the angle ‘dθ’ multiplied by the radius ‘r_i’, and is also equal to the path tread by the wheel, illustrated in Figure 13, equivalently expressed as the left wheel radius ‘w_i’ multiplied by the incremental angle turned by the wheel ‘dθ’. Assuming no wheel slip, then:

\[ ds_i = r_i \cdot d\beta = w_i \cdot d\theta, \]
\[ ds_r = r_r \cdot d\beta = w_r \cdot d\theta, \] (2.7.1)

The wheel path turning radii are related to the instantaneous turning radii ‘ρ’ according to

\[ r_i = \rho + l \]
\[ r_r = \rho - r \] (2.7.2)

Substituting Equation (2.7.2) into Equation (2.7.1) for \( r_i \) and recognizing that \( d\beta \) equals \( d\psi \) as stated above, the “F” equation can be solved for \( \rho \), yielding
\[
\rho = \frac{w_i \cdot d\theta_i}{d\psi} - l \tag{2.7.3}
\]

Substituting this expression for \( \rho \) in the "r" portion of Equation (2.7.2), then substituting the resulting expression for \( r \) back into Equation (2.7.1) and finally solving for \( d\psi \) gives

\[
d\psi = \frac{w_i \cdot d\theta_i - w_r \cdot d\theta_r}{l + r} \tag{2.7.4}
\]

Integrating from zero to the current time step gives an expression for vehicle heading, \( \psi_k \):

\[
\psi_k = \int_0^k d\psi = \int_0^k \left( \frac{w_i \cdot d\theta_i - w_r \cdot d\theta_r}{l + r} \right) = \frac{w_i \cdot \theta_{i,k} - w_r \cdot \theta_{r,k}}{l + r} \tag{2.7.5}
\]

where \( \theta_{r,k} \) and \( \theta_{i,k} \) are the respective wheel rotation angles recorded at time step \( k \), and all the initial conditions are zeros.

Return to expression (2.7.1). Since the arc length of the wheel path is linearly related to the incremental turning angle \( d\beta \) by the wheels turning radius \( (r_r \text{ or } r_l) \), the following linear interpolation is made to find the value of the distance traveled by the center of turning:

\[
ds = \frac{(ds_i - ds_r)}{(r_r - r_l)}(\rho - r_r) + ds_r \tag{2.7.6}
\]

Substituting Equation (2.7.2) into Equation (2.7.6) for \( r \) and \( r_l \) and simplifying gives

\[
ds = \frac{(ds_i - ds_r)}{(l + r)} \cdot r + ds_r \tag{2.7.7}
\]
and substituting for $ds_f$ and $ds_r$ from Equation (2.7.1) gives

$$ds = \frac{(w_l d\theta_l - w_r d\theta_r)}{(l+r)} \cdot r + w_r d\theta_r \tag{2.7.8}$$

Multiplying the second term by $(l + r)/(l + r)$ and combining terms we arrive at

$$ds = \frac{(w_l d\theta_l r + w_r d\theta_r l)}{(l+r)} \tag{2.7.9}$$

and finally integrating

$$s_k = \int_0^k ds = \frac{(w_l \theta_{l,k} r + w_r \theta_{r,k} l)}{(l+r)} \tag{2.7.10}$$

where $s_k$ is the linear distance traveled by the robot's center of turning at time step $k$. The method used for deriving global coordinates from this information is detailed in Section 4.4 - Map Making. Also included in Section 4.4 is an illustration of how these formulas were verified.
3 Current Hardware Development

The goal of the current development efforts, both in terms of hardware and software, has been to complete the integration of the MIMNS device into a robotic unit and to record substantial and verifiable data for exploring the effectiveness of the MIMNS hypothesis developed in Section 2.1. Because of this author’s familiarity with the Basic Stamp micro-controllers and their adaptability and accessibility, the decision was made early on to continue their use. Parallax Inc., the manufacture of the Basic Stamps, also offers a wide range of peripheral products designed specifically for integration into systems controlled by Basic Stamps. Due to limitations in processing speed, it was also decided to preserve the two-controller design, with one micro-controller acting as a dedicated data collector while the other handles environmental data and motion control. A hardware layout schematic is shown in Figure 15, with the various components described in the sections to follow.

3.1 Wheels & Encoders

Several changes were made to the mechanical design in an attempt to improve the functioning of the robot in its environment. Firstly, the rear wheels were removed and replaced by a single, supporting ball bearing caster. With only two drive wheels the robot can be differentially steered without the chattering described in Section 1. The wheel mounts and drive assemblies for the rear wheels were left in place however, and two US Digital E6S incremental shaft encoders (denoted EN1 and EN2 in the schematic) were installed in place of the wheels, allowing for comparison between collected encoder data (i.e. traditional “dead reckoned” data) and inertial position estimation techniques.
The optical encoders consist of a small translucent disk that attaches directly to the rotating shaft and contains 1024 evenly spaced dark lines around its perimeter, which gives the encoder a detection resolution of 1/1024 revolutions, or approximately 0.35°. The lines pass through a stationary optical detector that changes the state of its output according to whether or not it “sees” a line currently. As the shaft and attached disk rotate, the output of the encoder is a square wave corresponding to the passing of lines through the optical detector. In the case of the encoders selected for this unit, there are actually two optical detectors “watching” the disk, slightly offset from one another creating a phase shift between the two otherwise identical output signals used to determine the direction of rotation. The effect can been seen more clearly in Figure 14, where the two encoder outputs are labeled “CH A” and CH B.” Note that one channel leads the other depending on which direction the shaft is turning.

![PC6-83-X Timing Diagram](image-url)

**Figure 14 - PC6 Timing Diagram (PC6-83-1)**
Figure 15 - Hardware Schematic
PC6 encoder-to-counter converters, also from US Digital, (E2C in Figure 15) were installed inline between the encoders and the device used to capture encoder data. The input to these converters is directly from the encoders described above. The PC6 modifies the signals for use in “up/down” counters by comparing the two encoder signals, interpreting which direction the shaft is turning (i.e. “up” or “down,” in the language of the counter), and outputting single clock pulses on one channel for the up direction and on a second channel for the down direction. Again the effect is visualized more concretely in Figure 14 (an excerpt from the timing diagram for the PC6 datasheet from US Digital.) The converters installed on the robot are the PC6-83-1 type that output one “clock” pulse for every line that passes through the encoder’s optical eye, illustrated in Figure 14.

In order to capture and interpret data from the encoders, a PWMPAL peripheral device from Parallax was installed in parallel with the data collection micro-controller (see the PWMPAL in Figure 15.) The device features four 12-bit counters functioning in the background and requiring no active processing on the part of the controller except for setup instructions and the transmission of the current count, done at the controller’s discretion. Every “up clock” pulse from the right encoder is counted on one counter, every “up clock” pulse from the left encoder on another counter, etc. Each “up” pulse corresponds to 1/1024 of a revolution by the corresponding wheel in the forward direction, and each down clock pulse to 1/1024 of a revolution in the reverse direction. To determine the displacement of each wheel, the micro-controller subtracts the total number of down counts from the total number of up counts and multiplies by 1/1024.
times the wheel circumference. These distances are then stored in memory for later use in creating a map of the path traveled.

The standard issue PC6 encoder-to-counter device has output clock pulses nominally six microseconds in width. Upon experimentation it was found that this pulse width was in fact too narrow for reliable detection by the PWMPAL's counters. Utilizing the Basic Stamp's "PULSOUT" command to test the detection of varying pulse widths, it was discovered that a minimum width of 15-µs was required for reliable capture of all emitted pulses. Decreasing the pulse width below that value results in a loss of data, such that the PWMPAL was typically only able to detect about 60-70% of the six microsecond pulses emitted by the PC6 converter. This problem was remedied by replacing the standard PC6 with a modified version available from US Digital that has a 50-µs nominal pulse width, which allowed the PWMPAL to correctly identify pulses as intended.

3.2 Motion Control

The existing H-bridge motor controllers along with their supporting hardware were discarded in favor of a "Motor Mind C" motor controller developed by Solutions Cubed and distributed by Parallax. This controller interprets synchronous serial commands from the Basic Stamp micro-controller and outputs a pulse width modulated DC motor control signal. The output signal is a series of positive voltage pulses on the motor coils, or alternatively, a continuous square wave varying between ground and positive voltage. Figure 16 illustrates a typical PWM signal for a 12-volt DC motor. The term "duty cycle" is defined as the proportion of each period of the square wave given in the positive or "on" state. By increasing or decreasing the duty cycle (i.e. pulsewidth) of the constant
frequency square wave output, the Motor Mind C controller creates the same effect on a
motor as though it were increasing or decreasing a continuous analog voltage on the
motor coils, thereby affecting the torque output of the motor.

![PWM signal diagram](image)

**Figure 16 - PWM signal**

The Motor Mind C Carrier Board is another product offered by Parallax which integrates
the MMC controller with any BS2 type Basic Stamp micro-controller on a single printed
circuit board. The carrier board also contains screw terminals for motor and battery
connections (M1, M2, B1 and B2 in Figure 15) as well as terminals for direct access to
micro-controller pins, making it an indispensable medium for system integration. The
equipment and connections encapsulated in the carrier board appear within the dashed
boundary of Figure 15, and can also be seen in Figure 17, where it is the blue collared
circuit board on the right hand side of the image. Figure 18 provides a layout of the
major components of the robot as they appear in the image of Figure 17.

Also, as shown in Figure 15, a wire is used to connect pin 9 on the MMC to pin 5 on the
data control stamp. This connection allows the motion control stamp to signal to the data
collection stamp when the robot is not moving.
Figure 17 - View of the Current Robot

Figure 18 – Layout Schematic of Current Robot
3.3 Environmental Sensing

Two additional ultrasonic sensors were added to the existing environmental sensing platform, both facing forward at ±45° angles from the centerline of the chassis, bringing the total number of sensors to five. All five sensors were also upgraded from the existing Devantech SRF04 rangefinders to “PING” ultrasonic rangefinders from Parallax. The new generation of sensors achieves the same performance characteristics, measuring distances to objects from 3-cm to 3-m in distance, while using only a single Input/Output (I/O) pin on the micro-controller per sensor. A significant advantage is realized over the Devantech model, which required separate input and output pins for each sensor.

Besides improving the effectiveness of the simple obstacle detection and avoidance routines, the two additional sensors are used in a more intelligent hallway navigation algorithm. If the robot is offset from the hallway center or has a heading directing it away from center, the two diagonal-looking ultrasonic sensors will return distinct distances to the left and right walls, as is illustrated in Figure 19. In the figure “L₁” and “Lᵣ” are the left side and right side ultrasonic ranges while “L” is the hypotenuse of the triangle formed by L₁ and Lᵣ. The correction angle (β in Figure 20) is the angle between the robots current heading and the desired heading, which defines the direction of the path that gradually returns back to the centerline of the hallway at the point where “L” meets the hallway centerline.
Figure 19 - Correction angle diagram

Figure 20 - Correction Angle Derivation
To determine $\beta$, first a line is drawn from the midpoint of $L$ to meet $L_i$ at a right angle, this line is shaded gray in Figure 20. Using basic trigonometric definitions, the length of this line is $L/2 \cdot \sin(\phi)$. If the unknown length of the red shaded "Desired Path" shown in Figure 19 is $x$ then the Law of Sines states that

$$\frac{L/2 \cdot \sin(\phi)}{\sin(\theta)} = \frac{x}{\sin(90^\circ)} = x \quad (3.3.1)$$

Because of the $\pm 45^\circ$ orientation of the ultrasonic sensors, lines $L_i$ and $L_r$ intersect at a right angle, therefore, once again using the Law of Sines,

$$\frac{L}{\sin(90)} = \frac{L_r}{\sin(\phi)} \quad (3.3.2)$$

Also

$$\sin(\phi) = \frac{L_r}{L} \quad (3.3.3)$$

and

$$\phi = \sin^{-1}\left(\frac{L_r}{L}\right) \quad (3.3.4)$$

Using the Law of Cosines to solve for $x$ we arrive at

$$x = \sqrt{L_i^2 + (L/2)^2 - L \cdot L_i \cdot \cos(\phi)} \quad (3.3.5)$$

Substituting for $\phi$ and $x$ in Equation (3.3.1) and rearranging gives
\[
\sin(\theta) = \frac{\frac{L}{2} \cdot \frac{L_c}{L}}{\sqrt{L_i^2 + (L/2)^2 - L \cdot L_i \cdot \cos(\phi)}}
\]  
(3.3.6)

Simplifying and solving for \( \theta \) yields

\[
\theta = \sin^{-1} \left( \frac{L_c}{2 \sqrt{L_i^2 + \left(\frac{L}{2}\right)^2 - L \cdot L_i \cdot \cos\left(\sin^{-1}\left(\frac{L_c}{L}\right)\right)}} \right)
\]  
(3.3.7)

Finally, by inspection, the correction angle \( \beta \) is \( \theta \) minus 45°. Unfortunately the Basic Stamp is not capable of performing the necessary floating-point arithmetic or inverse trigonometric functions required to calculate \( \theta \) by itself. To aid the Stamp in this process, a floating-point coprocessor from Micro-Omega was installed into the system, and is labeled "uM-FPU" in Figure 15. The coprocessor accepts inputs and commands in synchronous serial formats, and the algorithm developed for this application returns the angle \( \theta \) according to Equation (3.3.7), in degrees, for the Basic Stamp to further process.

The stamp is programmed to recognize that if \( \theta \) is greater than 45° the robot must turn to the right, while if \( \theta \) is less than 45° the robot must turn left. Based on this logic, the stamp issues commands to the Motor Mind C so that the robot will gradually correct it's path heading towards the center of the hall at a rate proportional to the magnitude of the correction angle; i.e., the more correction required, the sharper the induced turn.

In the event that the robot enters an abrupt corner in the hallway or finds itself in a dead end, it will first come to a stop and then use the side facing ultrasonic sensors to determine if there is a direction out of the current end path (as in the case of a corner in
the hallway). The robot then rotates in place until facing the direction of its new open path and proceeds as normal.

3.4 **MIMNS Data Collection**

The existing yaw rate gyro installed on the robot was not functioning correctly, although the cause of these malfunctions was unknown. One suspected source of error was in the complex mounting mechanism of the instrument which is a 32-BGA (Ball Grid Array) type, requiring special equipment to install. Rather than leads or wires mounted into or completely through the PCB, a BGA arrangement has a bottom-side grid of small hemispherical metallic protrusions that function as electronic leads. This is one of many new styles of PCB surface mount technology designed to conserve physical space in an ever shrinking electronic environment. Inspection of an unpopulated version 2.0 MIMNS PCB revealed that several of the BGA leads on an installed gyro were not properly aligned to the solder pad for creating an electrical connection to the PCB. The missed connections appeared to be redundant, with the required connection supposedly taking place at another “ball” in the array, but it was suspected that this oversight might have been directly or indirectly causing the gyro malfunction.

To remedy this problem, the ADXRS150EB development kit was selected to replace the XRS150 on a new PCB design. The XRS150EB was created by Analog Devices as an evaluation board for customers considering using the XRS150 rate gyro in their projects. The board alleviates the need for the user to perform BGA soldering by providing a pre-installed rate gyro with all necessary signals routed to a 20-pin DIP (Dual Inline Package)
interface, which is much simpler to use and install. The board also contains all of the peripheral circuitry and components required, enabling the gyro to function correctly, so in essence the new sensor is a ready-to-use device, requiring only the addition of an analog to digital converter (ADC) for sampling by the micro-controller. The ADC used in these circuits is an LTC1298 from Linear Technologies, having a 12-bit resolution and synchronous serial output for simple interfacing with a micro-controller, primarily with the Basic Stamps.

The MIMNS PCB layout was altered to accommodate the 20-pin DIP interface and features were also added to improve interfacing with the MIMNS cube superstructure. The board was resized to fit into the precision-machined surfaces of the cube, and all components were moved to be within the 1.0-in square thru-holes labeled in Figure 10, eliminating mechanical interference from protruding electrical leads. Plated thru-holes for mounting screws were also added into the design, taking advantage of the precision alignments afforded by the cube design. The board was designed using ExpressPCB version 5.1.1 and ExpressSCH version 5.1.1 from ExpressPCB. The physical layout of the current design is shown in Figure 21, while the wiring schematic is shown in Figure 23. In the representation of the physical layout, yellow represents a silk screen pattern and indicates the location of the gyro chip (U1), accelerometer (U2), and ADC for measuring the gyro output (U3). Red and Green represent traces on the top and bottom layers of the board respectively, and any annular rings represent plated through holes which are used either to connect traces between layers or (in the case of the larger, unconnected rings) as clearance holes for mounting hardware.
To test if the new gyros were functioning properly once installed, the robot was rotated by hand 90° in the counter-clockwise direction, followed by a pause and a rotation back to the original orientation. These rotations were accomplished by hand during a series of tests, and were therefore an approximation with an expected ±10° error. The section of rotational rate data recorded from the gyro was isolated and then integrated over time for estimating the measured orientation angle. Figure 22 shows the results of this experiment, and the angle estimation is close to what was expected, with the robot
turning approximately 100° in the counter clockwise direction and returning to within a few degrees of its original orientation at time $t_o$. 

Figure 22 - Rate Gyro Test
Figure 23 - PCB Schematic
3.5 Onboard Software

Previous versions of the onboard software were designed to encompass running any of several unrelated tasks depending on the state of input pins which were set by the user prior to start up. For example, by connecting pin 13 of the data collection stamp to ground, (i.e. ‘setting pin 13 low’) the user signified that the robot should proceed with the data gathering routine, whereas if the pin were set high, the micro-controller would begin exporting stored data via its serial terminal. The limitation to this method was the loss of the I/O pins for any further functionality once the robot had been started. The decision was made to break out disparate tasks into separate programs, uploaded at the time of their intended use, thus freeing program space, RAM, and extra pins connections.

The stamp handling environmental sensing and motion control requires one program that is unmodified between runs. The programming procedure is illustrated in a flow chart format shown in Figure 24. Upon startup, the stamp initializes a counter, which is incremented every time a sense-move cycle is completed. When this counter reaches a user-defined threshold (e.g., 20 cycles) a trigger is initiated by the motion control Stamp unit stopping the robot, a “stopped” signal is sent to the data storage micro-controller, and finally, an appropriate length of time is allotted the data collector to re-initialize before reentering the sense-move cycle and restarting the counter. Analyzing the data during these stopped periods, the two dimensional assumption made in Section 2.2 is used to “zero out” the sensors, i.e., to calculate a new static bias. Chapter 5 makes evident the need for such a technique, since anomalies in the accelerometers caused dramatic and unpredictable changes in the bias calculation between static periods.
Except for the periodic stopping described above, the motion control program runs in a continuous loop, the first step being to poll each of the five ultrasonic sensors repeatedly and compare readings to ensure against faulty data. If the forward path is found to be blocked, the controller brings the robot to a stop and then analyzes the inputs of the left and right side sensors, deciding which direction to turn based upon which side appears to have more free space. The robot then instructs the Motor Mind C to set motor duty cycles initiating a timed turn in the direction of choice and then returns to the top of the sense move cycle. If the forward path is not blocked, the controller uses the \( \mu \)M-FPU math coprocessor to calculate the correction angle \( \theta \) as shown in figure Figure 20, and proceeds to set the motor duty cycles, steering the robot gradually back towards the center of sensed space.

The data collection algorithms, which were originally one encapsulating read/write program, have been separated so that one program is used to collect the data and another program is used to retrieve the data from memory and prepare the memory for the next round of data gathering. The data collection routines have been further divided depending on the type of data being taken in the particular trial, since it was found upon experimentation that if all of the inertial sensors and wheel encoders were polled and written to memory during each cycle, the sampling rate was an unacceptably slow 1 sample per second using a BS2IC Basic Stamp.
When the processor was upgraded to a BS2pxIC Basic Stamp, which executes program instructions approximately 4.5 faster than the BS2IC, the sampling rate improved to approximately 3 samples per second; however, upon further testing with the post processing routines, an increased sampling rate deemed desirable. If the robot is only
regularly polling a few necessary inertial sensors, the sampling rate can be improved to approximately 10 samples per second, which corresponds the time step value used in all the simulation and post processing routines (see Section 4). Thus, while the overall data collection routine remains the same, the programs are separated depending on which sensors are to be polled and recorded during the particular trial. The generalized structure of the data collection algorithm is shown in Figure 25.

![Figure 25 - Data collection flow chart](image)

For data extraction, the micro-controller accesses each location in memory sequentially and exports the data one byte at a time through a serial terminal to a PC. The data export format is comma delimited and samples are separated by a line feed so that a time dependant array of data is created. A typical sample of data will have several features that any post-processing software must be able to recognize. Firstly, when the data stamp initializes or is reset, it inserts a full line of zeros into the data as an indicator of what has
occurred. This line is typically followed by lines of data corresponding to information that is recorded while the robot is static. An example of a typical data extraction is shown below in Table 1:

<table>
<thead>
<tr>
<th>Data</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>Zeros indicating static period</td>
</tr>
<tr>
<td>1974 33 34</td>
<td>Average Gyro, Accelerometer “T2” for x1, x2</td>
</tr>
<tr>
<td>1977 5328 5787</td>
<td>Raw data for gyro, x1,x2</td>
</tr>
<tr>
<td>1972 5325 5782</td>
<td>Raw data for gyro, x1,x2</td>
</tr>
<tr>
<td>1977 5320 5784</td>
<td>Raw data for gyro, x1,x2</td>
</tr>
</tbody>
</table>

Table 1 – Exported data example

By contrast, if the data-recording stamp is recording encoder values, the data will consist of four columns of data, two for each encoder. Since the stamp is only capable of handling “words” of data, 16-bits in length, in order to transfer the 32-bit long encoder count for each wheel, it must break it into two words (a “high” and a “low” word) and transmit those words separately, recombining them later in the post processing routine. Since the encoders do not suffer from the signal biases that are inherent to the inertial instruments, there is no need for the stamp to collect any “static” information for the encoders, thus simplifying the export format to four columns of sequential data without the indicator lines of Table 1.
4 Offline Software Development

It was discovered through the course of practical application that simply integrating the MIMNS equations as described in Equation (2.1.16) and feeding them back into Equation (2.1.14) or (2.1.15) caused the system to become unstable, rapidly diverging from the expected value at an exponential rate. Alternate methods for filtering the MIMNS information were sought, and an extended Kalman filter was designed and implemented based on the volume by Zarchan and Musoff, [27] to serve this purpose.

4.1 MIMNS Kalman Filter

A Kalman filter is commonly referred to as an estimate-update or prediction-correction technique. The filter uses a system model to predict the estimated value of the system’s states one time step in the future and then compares that estimate to the measurements of the system’s states. At its heart, the Kalman filter uses the Riccati differential equations to derive what is referred to as the “Kalman gain,” a modifier which is applied to the residual (i.e. error) between what was predicted by a system model and what is observed during measurement. Intuitively speaking, the Kalman gain effectually weights either the estimate or the measurements according to which it decides is closer to the true state.

The first step to developing a Kalman filter is to develop a system model. The states to be estimated are the accelerations $a_{x1}$ and $a_{x2}$ sensed by the two accelerometers and the rotation rate “$r$” sensed by the rate gyro. Because the MIMNS robot does not record the deterministic control inputs (i.e. the motor control inputs), no knowledge of these inputs is available for post processing. In order to create a system model, a quasi-static
assumption was made for the states of the accelerations, stating that they do not change from instant to instant. This is expressed by the differential equations

\[
\begin{align*}
\ddot{x}_1 &= u_s \\
\ddot{x}_2 &= u_s
\end{align*}
\] (4.1.1)

where \( \ddot{x}_i \) represents the derivative of the acceleration terms and \( u_s \) is a zero-mean process noise, which is a time varying input of random gaussian noise. The actual value of \( u_s \) does not necessarily correspond to any particular physical equivalent, but rather reflects our uncertainty in the accuracy of the system model, essentially informing the filter that even though the system model states that the accelerations will not change from instant to instant, we believe some change will occur anyway. Though the quasi-static assumption does not truly reflect the dynamics of the system, it was hoped that the robustness of the Kalman filter would account for this lack of accuracy.

The model for the rate gyro is assumed directly from the two dimensional MIMNS Equation (2.2.2) and is expressed here as

\[
\dot{\rho} = -\frac{(\ddot{x}_{x,1} - \ddot{x}_{x,2})}{\ddot{y}_x} - \frac{\rho^2 \cdot \ddot{x}_x}{\ddot{y}_x}
\] (4.1.2)

where \( \ddot{x}_{x,i} \) are the accelerations described by Equation (4.1.1). Because the second term is non-linear, an Extended Kalman Filter (EKF) must be used. An EKF utilizes a linearized system model for the calculation of the Kalman update, which requires a linear model in order to be implemented. Outside of the Kalman update however, the non-linear system model can be propagated using any numerical technique (e.g. Euler’s
method or a Runge-Kutta method) to ascertain the system model prediction. The state space model of the above equations will be of the form

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{r}
\end{bmatrix} = \begin{bmatrix} A \\ 0 \end{bmatrix} \begin{bmatrix} x_1 \\
 x_2 \\
r
\end{bmatrix} + \begin{bmatrix} \bar{w} \end{bmatrix}
\]

(4.1.3)

where \( \bar{w} = [u_s \ u_s \ 0]^T \) and \( A \) is the Jacobian of the non-linear system model given by

\[
A = \begin{bmatrix}
\frac{\partial \dddot{x}_1}{\partial x_1} & \frac{\partial \dddot{x}_1}{\partial x_2} & \frac{\partial \dddot{x}_1}{\partial r} \\
\frac{\partial \dddot{x}_2}{\partial x_1} & \frac{\partial \dddot{x}_2}{\partial x_2} & \frac{\partial \dddot{x}_2}{\partial r} \\
\frac{\partial \ddot{r}}{\partial x_1} & \frac{\partial \ddot{r}}{\partial x_2} & \frac{\partial \ddot{r}}{\partial r}
\end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\
0 & 0 & 0 \\
-1 & 1 & 2 \cdot r \cdot \dddot{x}_s \\
\bar{y}_s & \bar{y}_s & \bar{y}_s
\end{bmatrix}
\]

(4.1.4)

The value for \( A_{33} \) is calculated before each iteration using \( \ddot{r} \), i.e., the value of the Kalman estimate for \( r \) from the previous iteration. Next, the above continuous system model must be modified to account for discrete effects since it will be implemented on a digital computer, in a discrete time domain. To do this, the fundamental system matrix \( \Phi_k \) is calculated according to

\[
\Phi_k(t) = e^{At}
\]

(4.1.5)

which can also be expressed in the form of a Taylor series expansion as

\[
\Phi_k(t) = I + At + \frac{1}{2}(At)^2 + O(H)
\]

(4.1.6)

where \( I \) is an identity matrix and \( O(H) \) is the sum of the higher order terms of the infinite series. Zarchan and Musoff [27] have shown through practical application that the
Kalman filters typically function adequately using the zeroth and first order terms of the above expression are included, reducing the above expression to

$$\Phi_k(t) \approx I + At$$  \hfill (4.1.7)

Carrying out the calculations:

$$\Phi_k(t) \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\frac{t}{\bar{y}_x} & \frac{t}{\bar{y}_x} & -\frac{2rt \cdot \bar{x}_x}{\bar{y}_x} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{t}{\bar{y}_x} & \frac{t}{\bar{y}_x} & 1 - \frac{2rt \cdot \bar{x}_x}{\bar{y}_x} \end{bmatrix}$$  \hfill (4.1.8)

Substituting the sampling time $T$ into the equation for $t$ we arrive at the discrete system dynamics matrix $A_k$:

$$A_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{T}{\bar{y}_x} & \frac{T}{\bar{y}_x} & 1 - \frac{2rT \cdot \bar{x}_x}{\bar{y}_x} \end{bmatrix}$$  \hfill (4.1.9)

The actual Kalman filtering approach is considered next. In his seminal paper in 1960 and subsequent publications, R.E. Kalman laid the theoretical groundwork for a new optimal filtering and state estimation technique which ultimately requires the solving of a Riccatti differential equation [28,29]. Zarchan and Musoff make use of the following recursive method for solving the equation, which they refer to as the “matrix Riccati equations,” expressed in notation convenient for a Kalman filter as shown in [27], i.e.,
The three matrices in Equation (4.1.10) are calculated at each time step, with state covariance matrix $P_k$ being propagated forward to the next iteration as $P_{k-1}$. The Kalman gain, $K_k$, will be applied at a later step to the system residual. The discrete system dynamics matrix, $A_k$, is already derived, and $Q_k$, $H$ and $R$ are to be defined.

The continuous process noise matrix $Q_s$ is related to the process noise vector $w$ shown Equation (4.1.3) by the statistical expected value function $E$ as follows

$$Q_s = E(ww^T) = \begin{bmatrix} \Phi_s & 0 & 0 \\ 0 & \Phi_s & 0 \\ 0 & 0 & 0 \end{bmatrix} = \Phi_s \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

(4.1.11)

where $\Phi_s$, like $\nu_s$, is a constant that does not necessarily correspond to any physical value, but rather reflects our uncertainty about the accuracy of the system model. Thus, if we believed the system model to be a perfectly accurate representation of the real world, $\Phi_s$ can be set to zero, whereas if we knew that our model had inaccuracies built in we could increase the value of $\Phi_s$, effectively “letting the filter know” that the model was not perfect. The numerical value of $\Phi_s$ is determined experimentally, and as Zarchan and Musoff have shown, it can often be used as a sort of “fudge factor” to help make the filter more robust [27].
The continuous process noise matrix, $Q_t$, must now be discretized to define $Q_k$, which is accomplished according to

$$Q_k = \int_0^T \Phi_k(\tau) \cdot Q_t \cdot \Phi_k(\tau)^T d\tau \quad (4.1.12)$$

where $\Phi_k(t)$ is the fundamental matrix of Equation (4.1.8). Carrying out the multiplication and the integration we arrive at

$$Q_k = \Phi_k \begin{bmatrix} T & 0 & -\frac{T^2}{2 \cdot \bar{y}_x} \\ 0 & T & \frac{T^2}{2 \cdot \bar{y}_x} \\ -\frac{T^2}{2 \cdot \bar{y}_x} & \frac{T^2}{2 \cdot \bar{y}_x} & \frac{2T^3}{3 \cdot \bar{y}_x^2} \end{bmatrix} \quad (4.1.13)$$

To define $H$ and $R$ we must first develop an estimation model for our sensor inputs. It is assumed that the sensor inputs and the system states are linearly related to according to

$$\begin{align*}
\dot{x}_1 &= \bar{x}_1 + \nu_{x1} \\
\dot{x}_2 &= \bar{x}_2 + \nu_{x2} \\
r &= r + \nu_r
\end{align*} \quad (4.1.14)$$

where $\bar{x}_1$, $\bar{x}_2$ and $r$ are the measurements and $\nu_j$ are time dependant sensor noises.

Rewritten in state-space format, Equations (4.1.14) will be of the form
\[
\begin{bmatrix}
\bar{z}
\end{bmatrix} = \begin{bmatrix}
H
\end{bmatrix} \begin{bmatrix}
\bar{x}_1 \\
\bar{x}_2 \\
r
\end{bmatrix} + \begin{bmatrix}
\bar{\nu}
\end{bmatrix} 
\]  
(4.1.15)

where \( \bar{z} \) is the vector of measurements, \( H \) is the measurement matrix and \( \bar{\nu} \) is the measurement noise given by

\[
\begin{bmatrix}
\bar{z} \\
\bar{x}_{11} \\
\bar{x}_{12} \\
r
\end{bmatrix}, \quad
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}, \quad
\begin{bmatrix}
\nu_{x1} \\
\nu_{x2} \\
\nu_r
\end{bmatrix}
\]  
(4.1.16)

The measurement noise matrix \( R \) is related to \( \nu \) by the statistical expected value function \( E \) as follows:

\[
R = E(\bar{\nu} \cdot \bar{\nu}^T) =
\begin{bmatrix}
\sigma_{x1}^2 & 0 & 0 \\
0 & \sigma_{x2}^2 & 0 \\
0 & 0 & \sigma_r^2
\end{bmatrix}
\]  
(4.1.17)

where \( \sigma_{x1}^2, \sigma_{x2}^2 \) and \( \sigma_r^2 \) are the variances of the x-accelerometer noises and gyro noise, respectively. Estimates for these values are derived and verified in Section 4.2.

With all elements of the matrix Riccati equations defined, the filter can now be implemented. The Extended Kalman Filter equation follows the form

\[
\tilde{x}_k = \tilde{x}_k + K_k \left[ \bar{z}_k - H(\tilde{x}_k) \right]
\]  
(4.1.18)

The Kalman estimate at time step \( k \) (\( \tilde{x}_k \)) is equivalent to the system model prediction of the current state (\( \bar{x}_k \)) plus the Kalman gain multiplied by a residual term inside the square
brackets. The residual is the difference between the sensor measurements \((\bar{z}_k)\) and the predicted value of the sensor measurements according to the system model \((H(\bar{x}_k))\). The Kalman gains \((K_k)\), calculated in the second step of the Riccati formulations of Equation (4.1.10), are a matrix of weighting factors, which essentially place emphasis either on the system model, in the case where the gains are close to zero, or on the measurements, in the case where the gains increase towards unity.

### 4.2 Simulation

To test the filter, a simulation of the robot’s movement was created using MATLAB. The .m file for this simulation is shown in Appendix A. A time history of the simulated path and heading of the robot’s center of turning, which is a point located directly between the drive wheels, was developed as a truth model for the robot’s motion. For example, in the path illustrated in the following sections, the robot moves forward approximately 50-ft, stops, turns to the right 90-degrees, and accelerates forward until the end of the simulation. Once this has been established, the motion is transformed to the locations of the instruments using the relative acceleration Equation (2.1.6). The truth model was sampled at a rate of 10-hz, so that the sampling time \(T\) is 0.1-seconds.

In order to realistically simulate measurements, sensor noise was created using MATLAB’s WGN (White Gaussian Noise) command, which was then added to the sampled truth model. To derive an estimated noise power, the accelerometer and gyro datasheets from Analog Devices were first referenced [12,25] for the typical values of spectral noise density, which were \(200 \mu g \sqrt{Hz\ rms}\) for the accelerometers and
0.05 \( \frac{\text{deg}}{s} \sqrt{\text{Hz}} \text{ rms} \) for the rate gyro. To convert these values to noise power they must be multiplied by the bandwidth of the instrument outputs, which are set in both types of instruments by a simple RC circuit in which the designer sets the low pass filter frequency by choosing a capacitance. In the accelerometers' case, the bandwidth is set at 50-Hz, and the noise power therefore is 28.3-\( \mu \text{g} \) or, with 1-g being 32.2 ft/s\(^2\), 9.1126*10\(^{-4}\) ft/s\(^2\). For the rate gyro the output bandwidth is set to a 40-Hz, and the noise power is .008 deg/s or 1.3789*10\(^{-4}\) rad/s. Figure 26 a shows steady state simulated noise overlaid against the actual steady state data taken from each instrument. We can see that the predicted noise values are acceptably close to the real noise values for the purposes of simulation.
Finally, the simulated gyro measurements were arbitrarily offset by a bias of .1-rad/s. State estimates were propagated using a 2\textsuperscript{nd} order Runge-Kutta method as described in [27]. Figure 27 shows the Kalman filtered data along with the truth model (which we are hoping to approximate) and the measured gyro readings. Immediately we can see that even though the filter has been initialized properly (with all states set to zero) the filter quickly approaches and tracks the measured readings. Upon further consideration, it was determined that the cause of this behavior was a faulty assumption made in the estimation model of the sensors.
In the expression for the gyro measurement in Equation (4.1.14) there is no term to account for the bias offset, and thus the filter is ignorant of the possible existence of such a term, which explains why it quickly approaches the instrument values. To account for this effect, Equation (4.1.14) was modified such that

\[ r = r + v_r + B \]  \hspace{1cm} (4.2.1)

where \( B \) is the gyro’s bias; however, in order for the Kalman filter to function properly, the measurements must be linearly related to the states of the system model. To overcome this, the bias itself was treated as a state for the Kalman filter to estimate, modifying equation (4.1.15) as
\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
r \\
B
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
r \\
B
\end{bmatrix} +
\begin{bmatrix}
u_{x1} \\
u_{x2} \\
u_r \\
u_r
\end{bmatrix}
\] (4.2.2)

The measurement noise term for the bias was set to \(v_r\), the same value as the measurement noise on the gyro since the two terms are closely related so that they can be expected to vary in the same manner. The noise term is somewhat arbitrary, but the addition of bias noise in the filter increases robustness to unexpected changes, which is important given the quasi-static assumptions that were made while developing the filter.

### 4.3 Updated Kalman Filter

To model the gyro bias a quasi-static assumption was considered, since changes in bias over time are expected to be small. Expressing the assumption in differential equation format results in

\[
\dot{B} = u_s
\] (4.3.1)

where \(u_s\) is a time dependant process noise. Modifying Equation (4.1.4) to include this change yields
\[
A = \begin{bmatrix}
\frac{\partial \tilde{x}_1}{\partial x_1} & \frac{\partial \tilde{x}_1}{\partial x_2} & \frac{\partial \tilde{x}_1}{\partial r} & \frac{\partial \tilde{x}_1}{\partial B} \\
\frac{\partial \tilde{x}_2}{\partial x_1} & \frac{\partial \tilde{x}_2}{\partial x_2} & \frac{\partial \tilde{x}_2}{\partial r} & \frac{\partial \tilde{x}_2}{\partial B} \\
\frac{\partial \tilde{r}}{\partial x_1} & \frac{\partial \tilde{r}}{\partial x_2} & \frac{\partial \tilde{r}}{\partial r} & \frac{\partial \tilde{r}}{\partial B} \\
\frac{\partial \tilde{B}}{\partial x_1} & \frac{\partial \tilde{B}}{\partial x_2} & \frac{\partial \tilde{B}}{\partial r} & \frac{\partial \tilde{B}}{\partial B} 
\end{bmatrix}
= \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
-1 & 1 & -\frac{2 \cdot r \cdot \bar{x}_x}{\bar{y}_x} & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
(4.3.2)
\]

with the process noise vector \( \tilde{w} = [u_s \ u_s \ 0 \ u_s] \). Following the same process to derive \( A_k \) gives

\[
A_k = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
-\frac{T}{\bar{y}_x} & \frac{T}{\bar{y}_x} & 1 - \frac{2r \cdot \bar{x}_x}{\bar{y}_x} & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
(4.3.3)
\]

The continuous process noise matrix now becomes

\[
Q_s = E(ww^T) = \begin{bmatrix}
\Phi_s & 0 & 0 & 0 \\
0 & \Phi_s & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & \Phi_s
\end{bmatrix} = \Phi_s
(4.3.4)
\]

which is discretized according to Equation (4.1.12) to give

\[
Q_k = \Phi_s \begin{bmatrix}
T & 0 & -\frac{T^2}{2 \cdot \bar{y}_x} & 0 \\
0 & T & \frac{T^2}{2 \cdot \bar{y}_x} & 0 \\
-\frac{T^2}{2 \cdot \bar{y}_x} & \frac{T^2}{2 \cdot \bar{y}_x} & \frac{2T^3}{3 \cdot \bar{y}_x^2} & 0 \\
0 & 0 & 0 & T
\end{bmatrix}
(4.3.5)
\]

73
From Equation (4.2.2) the measurement matrix $H$ and measurement noise matrix $R$ are

$$
H = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix}, \quad R = \begin{bmatrix}
\sigma_{x_1}^2 & 0 & 0 & 0 \\
0 & \sigma_{x_2}^2 & 0 & 0 \\
0 & 0 & \sigma_r^2 & 0 \\
0 & 0 & 0 & \sigma_r^2
\end{bmatrix}
$$

(4.3.6)

The problem of supplying a “measurement” value of the bias ($\dot{B}$) for the Kalman filter to use in its calculation of the residual in Equation (4.1.18) is considered by assuming the estimator is “perfect” under ideal conditions, and defining the bias as the difference between the gyro measurement and the output of the Kalman filter. These values from the previous time step can be propagated forward and used to calculate a “measurement” for the bias.

$$
\dot{B}_k = r_{k-1} - \hat{r}_{k-1}
$$

(4.3.7)

Appendix B (Section 8.2) contains the updated .m-file MATLAB script used to simulate the above filter using the same truth inputs from Section 4.2. To test the robustness of this new filter, the initial estimate of the bias was deliberately offset to .2-rad/s, while the bias on the rate gyro was maintained at positive .1-rad/s. Figure 28 is an example of the results of these simulations, and the improvement is significant. Even with the filter’s initial estimate for $r$ significantly off due to an erroneously initialized bias term, the filter very quickly tracked near to the true value. There is also a “random walk” component to the error, which is due to the integration of the Gaussian noise. In the simulation shown in Figure 28 this error is relatively well behaved, i.e., it tends around the truth value; however, multiple trials show that a truly random behavior is observed, and may sometimes depart rapidly from the truth model. Experiments with the simulation
parameters showed that reducing the sampling time ($T$) or increasing the $y$-distance ($\bar{y}_x$) between the two simulated accelerometers (see Section 5.3 for discussion) both improved this aspect of the simulation, but the parameters used to generate Figure 28 most closely emulate what is expected for the robot, i.e., a sampling time of .1-seconds and a $y$ distance of 15-inches (again, see Section 5.3 for discussion of this distance value.)

![Simulated Rotation Rate $r$](image)

**Figure 28 - Updated Kalman Filter Simulations**

Upon experimentation with real data recorded by the robot, it was found that due to the quasi-static assumption made during modeling, the updated filter displayed some undesirable behaviors. While the robot was at a standstill, the bias state tracked quickly to the true value; however, once the robot began to move, the bias term would begin to
fluctuate such that often the $r$ term would remain steady while the bias term tracked the turning of the robot. This side effect proved to be unpredictable. A better result was achieved by using the bias-tracking Kalman filter to determine the bias during static periods and then to switch back to the original non-bias-tracking filter for the remainder of the time, and subtracting the estimated bias from the measurements, essentially rearranging Equation (4.2.1) to $r - B = r + v_r$. As described in Section 5.4, while functional results were achievable using this method, the filter essentially becomes a computationally expensive method of estimating the bias during static periods, which can be found more efficiently by averaging a limited number of the raw measurements from any static period.

In order to implement the change, the program switches to the bias-tracking filter and re-initializes whenever a row of zeros is encountered in the data (see Table 1), and then reverts to the original filter once the bias has been determined. The question then became how long to allow the bias-tracking filter to run to obtain a good estimate for the bias. According to Zarchan and Musoff [27], one way to determine the accuracy of the Kalman estimate is to study the time history of the diagonals of the state covariance matrix, $P$, which can be interpreted as the variances of the of the estimated error in the states. The square root of this time history is then the theoretical standard deviation of the Kalman estimate from the true value.

Figure 29 shows a plot of this standard deviation for the first 1.5 seconds of real data taken by the robot during a test run. Each of the $P$ diagonals is initialized at 10,000,
representing a high uncertainty about the initial conditions of each state. The bias standard deviation reduces dramatically and converges to a constant after just three or four time steps, equivalent to 0.3 or 0.4 seconds. The term converges on a non-zero constant due to the process noise defined by the matrix $Q$. If the process noise were set to zero (i.e. $\Phi_z = 0$), the diagonals of $P$ would eventually converge towards zero, meaning perfect certainty in the estimate, but since the system model is not perfect, as discussed in Section 4.1, process noise is necessary for providing sufficient filter robustness to compensate for imperfections. Taking this knowledge into consideration, the post-processing routine switches from the bias-tracking to the non-bias-tracking filter after 10 time steps, or 1.0 seconds.

![Figure 29 – Theoretical Standard Deviation in Kalman Estimate](image)

77
4.4 Map Making

With the Kalman filter now providing estimates for rotation rate, a consideration on how these affect the global position estimate of the robot is made. To achieve this, the recorded accelerations must first be transformed to the robot's center of rotation according to Equation (2.1.6) using the Kalman estimate for $r$ and $\dot{r}$. The acceleration measurements are then integrated twice, estimating displacement ('$s$'), using an iterative 2nd order Runge-Kutta technique. To find the heading angle ('$\psi$') the Kalman filtered MIMNS rotation rate estimate is integrated using the same R-K method. The displacement and heading angle time histories are then converted at each time step to the global coordinate frame according to following iterative technique:

\[
\begin{align*}
X_k &= X_{k-1} + (s_k - s_{k-1})\sin(\psi_{k-1}) \\
Y_k &= Y_{k-1} + (s_k - s_{k-1})\cos(\psi_{k-1})
\end{align*}
\] (4.4.1)

where $X$ and $Y$ are the robot's coordinates in the global plane. Intuitively, these equations express that since the previous time step $(X_{k-1}, Y_{k-1})$ the robot has moved forward a given amount $(s_k - s_{k-1})$ while oriented at angle $\psi_{k-1}$ with respect to the global coordinate system. Figure 30 illustrates a typical simulated global map, with the true path displayed along with the Kalman estimate and the estimate made using only the raw measurements. The figure shows that even though the filtered MIMNS data is not perfect, it presents a significant improvement over the unfiltered data.
Figure 30 – Simulated Global Map

Similarly, the estimates for $s_k$ and $\psi_k$ from the encoder model of Section 2.7 can be used with Equations (4.4.1) to generate a dead reckoned map based on encoder data. To test that the encoders were functioning properly, real encoder data was acquired while the robot was hand-propelled through a series of maneuvers. The first test was to push the robot along a straight path for 10 feet. The encoder data was extracted and converted to a global map using the method just described, and the results are displayed in Figure 31. The map shows a path which is nearly a straight line, traveling forward about 9.8 feet. Part of the inaccuracies are to do the inexactitude of pushing the robot by hand, but the majority is attributed to a lack of precise calibration. However, the results are a positive verification of the equations developed Section 2.7.
Figure 31 - Encoder Map: Straight Line Test

To further test the encoders and demonstrate the map creation techniques, another trial was performed, this time with the robot hand-propelled about three sides of a 4’ x 8’ table with the resultant map shown in Figure 32. As in the last trial, the data is uncalibrated, meaning that the values for wheel diameter and wheel base length were roughly estimated; however, the encoders are able to recreate the shape and size of the path within acceptable tolerances. The largest sources of error occur during the sharp 90-degree turns, during which wheel slippage was observed, typical of dead reckoning position estimation techniques.
Figure 32 - Encoder trial map
5 Data Collection and Results

5.1 Troubleshooting

As an initial attempt, a run was attempted down a hallway of the 2nd floor of the James E. Gleason building was performed. During the trial the robot transversed approximately 50-ft down an empty hallway to the corner of the building, turned to the right 90° before continuing another 20-ft. The recorded data was extracted from the robot and inputted into the MATLAB post-processing routine. Inconsistencies in the post-processed data were immediately observed. The routine generates a series of plots, one of which shows the difference between the two x-accelerometers divided by the "$\frac{y}{x}$" distance between them. This corresponds to the first term of Equation (2.2.2), and is plotted alongside the Euler derivative of the measured gyro data ($\dot{r}$).

While it was not expected that the two signals would overlay perfectly, since the "x-diff" term is only a partial expression of the MIMNS equation, it was hoped that some correlation would be evident in the plot. Figure 33 shows the graphical results from this data analysis, and it is evident from visual inspection that any rotation information contained in the partial MIMNS equation has been obscured. Further investigation supported this conclusion, thus initiating a series of troubleshooting trials attempting to determine the source of what appeared to be a relatively large source of noise contained within the measurements.
Figure 33 - First Run

Note that in Figure 33 there are periods of relative inactivity, for instance between approximately 50 and 57 seconds. These periods correspond to the motion controller reverting robot to a complete stop, and signaling the data handler of this condition, and then remained still for a brief period while the data controller performed a sub-routine of “motionless” activities. The source of the noise appeared to only be active during periods of motion, which meant that it was likely linked to the motors in some manner, either electrically or mechanically.

To test this theory, the robot was raised so that its wheels no longer touched the ground and a new set of data was taken while the motors spun freely. Since the chassis (hence the accelerometers) was not traversing during this time period, ideally the measurement
would reflect only a small, constant noise which is inherent to the instruments. Figure 34 shows the raw accelerometer data from this trial. Again, it is easy to discern the periods when motors are moving or not moving, as the “noise” in the data increases by an order of magnitude when the motors are spinning freely, although the difference is less extreme (note the change in scale of the vertical axes.)

![Figure 34 - Motor Test 1](image)

5.1.1 Electrical

A process of elimination approach was undertaken to attempt to determine if the source of the noise was electrical, beginning with eliminating the common ground plane that had heretofore been shared by all power sources. Even though both the motion control and
data-acquisition systems had separate power supplies, each ground plane had been electrically linked so that signals passed between the two controllers could be referenced to this “common ground.” Having this link is a necessity for meaningful data transmission; however, having the sensitive data acquisition instruments linked directly to the high impedance electrical load of the motors might cause undesirable fluctuations in the reference voltage of the instruments during periods of heavy motor activity.

Since there was no longer a common ground shared by both micro-controllers, data sent over the link between them was now meaningless and unreliable. The link is illustrated in the hardware schematic of Figure 15 as the wire connecting pin 9 on the motion control stamp to pin 5 on the data acquisition stamp, and is necessary for the motion controller to communicate to the data handler when the robot is stationary. To reestablish the communication channel, an “optical de-coupler” was built, consisting of a Light Emitting Diode (LED) whose light is cast on a photo-sensitive resistor. Figure 35 shows the circuit diagram used to construct the de-coupler.
Figure 35 - Optical De-coupler

The left half of the circuit is a simple voltage divider, in which the variable resistance value of R₂ is set by the amount of light impinging upon it, and V_D and G_D refer to the positive voltage (typically 5-volts on the Basic Stamp) and ground voltage on the data controller. On the right hand side, the state of Pin 9 is set by the motor controller relative to G_M (motor ground), turning the LED on or off, thereby affecting R₂. Finally, the fixed resistance R₁ is chosen such that the state of Pin 5 is varied across the TTL threshold (approximately 1.4-volts relative to G_D) according to whether the LED is on or off. Thus, by controlling the functioning of the LED, the motion control Stamp can control the state of a pin on the data acquisition Stamp, and can be utilized for sending simple signals without the need for a common ground.

A trial similar to the one depicted in Figure 34 was recorded with the two electrical networks de-coupled, but unfortunately, little improvement was observed. Under a
suspicion that the rapid pulsing from the encoders might be causing the interference, the encoders were disconnected and another trial was run, but this suspicion also proved unfounded. With the all the known possibilities for direct electrical interference investigated, a new experiment was attempted to try and ascertain whether the data acquisition process was properly functioning. The motion control equipment was disconnected from the batteries, and the robot was transported manually about. The trial consisted of a short period of transverse motion followed by a period of turning with no lateral motion, and finally followed by another period of transverse motion.

Figure 36 shows the raw acceleration measurements recorded during this trial, and although the numerical values do not have much significance due to the lack of an initial stationary period for calculating accelerometer zeros, the trends show that the accelerometers are indeed functioning correctly. Intuitively we know that a human pace results in periodic acceleration and deceleration with every stride taken, and this is reflected in the cyclic nature of the recorded accelerations during the two periods of walking.
5.1.2 Mechanical

The primary suspicion for the source of the "noise" now fell upon mechanical interference and the possibility that vibration from the motors combined with the large gear-heads was affecting the accelerometer devices. It was first important to note how the circuitry surrounding the accelerometers was affecting their output and how this might lead to interference from vibration. Figure 37 is a functional block diagram from the Analog Devices datasheet, illustrating the accelerometer along with the required user defined circuitry, which appears outside of the bold border [25]. According to the datasheet, the user-defined capacitor values $C_x$ and $C_y$, in conjunction with the factory
trimmed resistances, labeled "RFILT" in the diagram, act as a low-pass filter on the accelerometers output. By selecting the capacitance values the designer chooses the bandwidth of the filter and can selectively reduce the effects of undesirable frequencies on the instrument’s output. During the trials that produced Figure 34 and Figure 36, accelerometer circuits designed by C. Mellars were in place [21], which utilize 0.1-μF capacitors on all filter pins. The datasheet lists this capacitance as corresponding to a nominal filter “3-dB bandwidth” of 50-hz.

![Accelerometer Functional Block Diagram](image)

**Figure 37 - Accelerometer Functional Block Diagram**

If the output filter is passing frequencies up to 50-Hz, Nyquist’s principle implies that the data would need to be sampled at twice this value, 100-Hz, in order to guarantee against aliasing [30]. During the above trials, the data acquisition stamp was sampling at its maximum rate of 10-Hz, not fast enough to capture an accurate portrait of the signal. To quickly test whether this was indeed the source of the noise problem, a 0.47-μF capacitor was soldered in parallel to the existing 0.1-μF on one of the XFILT pins. According to the Analog Devices datasheet, a 0.47-μF capacitor corresponds to a 3-dB bandwidth of 10-
hz, so that the total capacitance of 0.57-\(\mu\)F now temporarily placed on \(C_x\) would have a bandwidth smaller than 10-Hz. A trial was run with robot’s wheels raised off the ground as in Figure 34, but with the extra capacitance placed on \(C_x\) for the \(a_{x,2}\) accelerometer. Figure 38 shows the acceleration data recorded during this trial, and the improvement is striking, with nearly an order of magnitude reduction in the power of the noise.

![Figure 38 - Accel Data with Lowered Bandwidth](image)

**5.2 Testing**

Based upon the previous result, the \(a_{x,1}\) accelerometer was modified in a similar manner and another experiment was run. The motors were left turned off but the robot was manually pushed through a series of maneuvers. This approach allows for repeatable
testing so that the results could be directly compared between runs. The pattern chosen consisted of the following steps:

1. No movement for 10 seconds,
2. Accelerate forward, move 10-ft,
3. Without stopping, turn 90°-CCW,
4. Continue forward another 5-ft and decelerate to a stop, pause briefly
5. Turn in place 90°-CCW, pause, turn in place back to previous heading, 90°-CW,
6. Accelerate forward, move 5-ft,
7. Without stopping, turn 90°-CW,
8. Continue forward another 10-ft and decelerate to a stop.

Figure 39 shows the raw MIMNS data along with the derivative of the rate gyro measurements, similar as shown in Figure 33. While not a perfect match, we can now see that the MIMNS estimation for \( \dot{r} \) is similar in shape as the derivative of the gyro output, which is especially evident between 35 and 45 seconds, corresponding to step five from the listing above.
At this point a simple observation about the 2-D MIMNS Equation (2.2.2) was made that had a crucial impact on the quality of the results. The equation is repeated here for convenience:

$$\hat{r} = \frac{(a_{x,2} - a_{x,1})}{\frac{r^2 \cdot \bar{x}_x}{\bar{y}_x}}$$  \hspace{1cm} (2.2.2)

In this equation, the choice of $\bar{y}_x$ - the “y” distance between the two $x$-accelerometer - plays a critical role in the quality of the estimated results. The reasons for this are twofold: the first is that the greater $\bar{y}_x$ is, the greater the radius from the center of turning to the accelerometers will be, and in turn, the greater the measured accelerations will be.
This is analogous to a spinning Frisbee, where points further from the center of the disk undergo greater accelerations due to the rotation. The second reason $\bar{y}_x$ is crucial is that in the above expression for $\dot{r}$ it essentially acts like a gain on the $a_{x,2} - a_{x,1}$ ("x-diff") term. If $\bar{y}_x$ is small, the gain will be large, and will have the effect of greatly amplifying any noise in the recorded measurements; however, if $\bar{y}_x$ is relatively large, it will actually serve to attenuate the effects of noise.

Thus it is desirable to make $\bar{y}_x$ as large as possible because it will both increase the acceleration signals in a mechanical sense and attenuate the effects of electrical noise; however, the tradeoff to making $\bar{y}_x$ larger is that we risk violating the rigid body assumption discussed in Section 2.6. The farther the accelerometers are placed, the more acceleration they will sense relative to one another due to flexion of the body. Through the course of repairing and refurbishing the existing robot, including the chassis structure, it was observed that the chassis is actually particularly flexible due to the lack of structural reinforcement in the original design. Add to this that the thin sheet aluminum seemed to be unusually soft during machining (only the sharpest of drill bits would actually cut the metal as opposed to merely "pushing" it,) and we quickly become aware of the potential problems associated with moving the accelerometers off the polymer cube and onto the chassis in order to increase the $\bar{y}_x$ distance. However, for the sake of experimentation, just such a modification was made (see Figure 40).
The accelerometers were relocated to the outer walls of the robot and the 8-step manual motion routine described earlier in this section was repeated. Figure 41 shows the results of this trial. The impact of increasing $\bar{y}_x$ was significant, and appeared to have easily outweighed whatever costs due to loss of rigidity from abandoning the polymer cube structure that may have occurred. The second part of Figure 41 shows the data recorded during the turn-in-place maneuvers of step five, and the correlation between the measured gyro readings and the truncated MIMNS equation is excellent, even with several acknowledged sources of error. This is an important point that should be weighed heavily into the designs of any future researchers.
5.3 Further Improvements

With the source of the "noise" positively identified, it was deemed safe to replace the version 2.0 MIMNS PCB’s with the updated version discussed in Section 3.4, though given the recent developments, some changes were required to the components which would populate the boards. First, it was no longer necessary for any gyros and accelerometers to be occupying the same board since the layout on the robot would now consist of one accelerometer chip at each outside edge of the chassis and a single rate gyro located in the center. On the accelerometer boards, since the 10-μF power-supply-decoupling tantalum capacitor was no longer necessary for the ADC chip, it was replaced
with a .1-μF ceramic disk capacitor, more in keeping with the decoupling needs of the accelerometer chips themselves. Also, the updated design had originally called for the use of .1-μF capacitors on the C_x and C_y filter pins; however given the discussion in Section 5.1.2 it was evident that this was no longer correct. The ADXL202E data sheet was consulted and the following equation for selecting the proper capacitance was used.

\[
F_{-3dB} = \frac{5 \mu F}{C_{x,y}}
\]  

(5.3.1)

\(F_{-3dB}\) is the desired filter three-decibel bandwidth and \(C_{x,y}\) is the capacitance necessary to set that bandwidth. In the case of the MIMNS data collection, with the sampling rate of the micro-controller capped at 10-Hz, in order to comply with Nyquist’s principle the measured frequencies must be smaller than \(\frac{1}{2}\) the sampling rate. Thus, for an \(F_{-3dB}\) of 5-Hz, \(C_{x,y}\) must be 1.0-μF. These changes are highlighted in Figure 42.

With these corrections in place, the version 3.0 boards were populated and installed onto the robot. The movement pattern of Section 5.2 was repeated and the results are shown in Figure 43 and reflect continued improvement of the approximate MIMNS estimation with respect to the rate gyro derivative.
Figure 42 - Updated PCB Schematic

Figure 43 - Results From New PCBs
5.4 Final Implementation

In another round of testing, the robot was re-introduced to its intended environment, the halls of the James E. Gleason building in RIT’s Kate Gleason College of Engineering. Through the course of the trial, the robot moved forward approximately 50-ft, turned clockwise 90° and continued another 50-ft down the hallway, pausing periodically to reestablish sensor zeros, with the whole trial lasting approximately two minutes. The data collected during this run was observed to be very similar to the data shown in Figure 43, and shared several key features with it. Figure 44 shows a close view of two regions of the data, which together are representative of capabilities of the current system.

![Figure 44 - Measurement Details](image-url)
The first part of the figure illustrates the typical signal recorded while the robot is turning in place, as when it is confronted with a wall and is forced to stop and turn in measured pulses to the left or right. In these situations even the rough estimation for the MIMNS equation used in these plots shows an excellent correlation between derived prediction and the recorded data. The second part of the chart shows data that is typical of periods when the robot is making small corrective turning maneuvers while also traveling forward. Throughout these portions of the trial the correlation is less obvious, though arguably still present.

The data was then Kalman filtered as described throughout Chapter 4. The first attempt with the filter was made with the process noise term ($\Phi_p$) set to zero. What this tells the filter, essentially, is that we believe that the system model is a perfect representation of the real world. Since this is not actually the case, significant error was expected, and Figure 45 validates this prediction. Since the filter believed the system model to be perfect, a short period of time after each initialization it was paying virtually no attention to the instrument measurements. Figure 46 shows the filtered results with varying values of process noise, in order to show the effect that increasing this key parameter has on the output. Beginning with $\Phi_p$ equal to 0.1 we see that the estimate begins to take some shape, but is not exceptionally accurate. Increasing $\Phi_p$ by an order of magnitude to 1.0 changes the shape of the data but yields little noticeable improvement. The third part of Figure 46 shows $\Phi_p$ increased to 10 and the Kalman estimate more closely matches the rate gyro output, minus the bias.
Figure 45 - Kalman Estimate, Zero Process Noise

Figure 46 - Kalman Estimate with Varying Process Noise
The process noised was continually increased until a point where any further increase showed little benefit. The Kalman estimate with $\Phi_s$ set to 500 is shown in Figure 47.

![Rotation Rate $r$](image)

**Figure 47 - Final Kalman Estimate**

At first glance, these results appear to be rather attractive, with the Kalman estimate having a shape very similar to the rate gyro output, only shifted such that the instrument bias is no longer present. The astute observer should notice, however, that the high process noise essentially turns the Kalman filter into a computationally expensive method of subtracting out the static bias in the gyro data. While this is a necessary step, it could equally be accomplished with far less work by averaging the gyro bias during a known static period and subtracting it from the measurements during the dynamic periods, much the same as was done to the accelerometer measurement inputs to the filter. In order for
the Kalman filter to provide better estimates without the use of excessive process noise, improvements to the system model are required, the most straightforward of which would be the inclusion of deterministic control input data (i.e. motor control outputs from the \( \mu \)-controller). Regardless, the Kalman filtered rotation data shown in blue in Figure 47 was used to create a map in accordance with Section 4.4, and the results are shown in blue in Figure 48 along with the map created using the raw gyro outputs from the same trial (in red in both figures.)

![Global INS Map of RIT Hallway](image)

**Figure 48 - Global INS Map of RIT Hallway**

When compared with the approximate truth model for the trial, overlaid in green, the Kalman estimate shows a significant improvement over the unfiltered data; however, it is still uncorrelated with the truth model. The unfiltered data, whose uncompensated bias
offset causes the appearance of continuous rotation, is marked by the many loops made in
the calculated path. This rotation also causes artificial acceleration components, created
when the accelerometer data is computationally transformed to the robot's center of
turning according to Equation (2.1.6). Though it is not a perfect representation of truth,
the shape of the Kalman filtered map reflects several important points about the robot's
behavior. For example, though the robot begins the trial facing directly forward in the
center of the hall, because of varying hallway geometry it begins to weave back and forth
as it directs itself towards the center of the hallway, overshoots and then corrects again.
The general shape of the path is also somewhat preserved, with the robot moving
forward, turning approximately 90° in a clockwise motion and moving forward again,
although the length of these legs are obviously distorted.

Though these results are encouraging, they are far from a perfect representation of the
robot's path. In reality the machine moved forward approximately 50-ft before turning,
and then moved forward another 50-ft after having negotiated the turn, but the generated
map shows the robot moving forward about 180-ft in the first leg and another 115-ft in
the second. For comparison, Figure 49 shows a map generated from encoder data (see
Section 2.7) of a trial similar to the one from Figure 48. As previously mentioned, the
encoder model sees the most significant introduction of error during periods of sharp
turning. Observation during these periods revealed increased wheel slippage, a form of
"non-systematic" error commonly identified as one of the major shortcomings of dead-
reckoned position estimates (see section 2.7). By contrast to Figure 48, the encoder map
sees a greater drift in the orientation over time, but appears to be far more accurate in terms of scale, with the robot traversing approximately 40-ft on each leg of the trial.

Figure 49 - Encoder Map of RIT Hallway
5.4.1 Accelerometer Anomalies

To gain a further understanding, the raw accelerometer data and the velocity information obtained from integrating them were viewed in the unfiltered states. As shown in Figure 50, directly after startup the accelerometers function as expected before the robot begins to move, at which point an added forward component to the acceleration is observed that remained in addition to any sensed components, as though the initial forward thrust caused the instrument to “stick” slightly higher in value than the previously calculated zero. This phenomena was most evident at near 30-seconds and 65-seconds in the acceleration data, at points where the robot had come to a stop and was not accelerating,
but prior to the data processing routine had re-zeroed the accelerometer data (the re-zeroing occurs at points where the data indicates the robot had come to a stop, see Table 1 for clarification.) After the period of acceleration, a visible new bias appeared in the accelerometer data indicating a forward acceleration where none had been before. Also, a discontinuity in the velocity data was observed after reinitialization to zero and the zero-velocity signal.

As shown in Figure 50 the effect of this “sticking” action seemed to diminish with each successive burst of motion. In fact, on the fourth and final period of motion, the data indicated that the robot actually decelerated to a stop completely on its own with no re-zeroing required. This is, of course, the desired behavior, which, if present continuously from the beginning, would yield a more accurate map. The fact that the “sticking” anomaly seemed to relieve itself over time indicated that there was some unknown transient property to the accelerometers not being modeled in the simple instrument model of Section 4.1, that could be predicted and compensated for in a simple manner.

Unfortunately no information provided in the manufactures data sheet or application notes immediately revealed the cause of the phenomenon, so another trial was attempted to observe for a repeatable trend. The robot was pushed through the manual trial pattern of Section 5.2, followed by a 180° turn, and a repeat of the pattern, minus step five, such that the robot returned to its starting location. Figure 51 shows a map generated with the Kalman filtered data:
Figure 51 - Accel Anomaly Test Map

The results display similar characteristics as the map of Figure 48. The pattern at least resembles what we would expect to see; however, the scale of the acceleration varies dramatically, causing severe displacement anomalies. Figure 52 displays the raw acceleration captured by the accelerometers during the trial. The previous hypothesis that the “sticking” phenomena diminishes over time, is invalid, since in this trial the effect increases with time, with both negative and positive offsets occurring. The phenomena appears to affect both instruments equally, with an approximately (but not exactly) equal amount of new bias occurring in each signal after periods of movement. This is not necessarily an undesirability result for the rotation estimate developed in Section 2 since one signal is subtracted from the other, and therefore the biases in the absolute accelerations are cancelled, with only relative differences between the two signals remaining.
5.4.2 Alternate MIMNS Methods

A MATLAB `.m-file` and accompanying Simulink diagram (Appendix 8.4) were generated by importing the data from the first portion of Figure 52 to calculate the various forms of the MIMNS estimate according to Sections 2.2, 2.3, and 2.4, with the result shown in Figure 53, plotted along with the rate gyro measurements for comparison. Recall that “MIMNS Direct Feedback” refers to the method in which the ‘r’ input is derived from the integrated MIMNS output, while the “MIMNS Velocity” methods integrated accelerometer information first to attain velocities before applying the MIMNS principal. “Gyro Feedback” refers to using the rate gyro measurement for the ‘r’ input.

The most interesting and obvious feature on the chart is the rapid departure of the “direct feedback” method towards the end of the trial. This brings to light an interesting aspect
of this particular method of feedback, and one that makes it probably less suitable for real world application. If an error is somehow induced into the estimate for the rotation rate ‘\( r \),’ and is then followed by a period where no rotation is observed, and if the induced error is positive, then the \(-r^2\bar{x}_x/\bar{y}_x\) term will cause a negative \( \dot{r} \) and the error decreases over time, decaying approximately exponentially towards zero. However, if the induced error is negative, the minus r-squared term still causes a negative \( \dot{r} \), and drives the estimate for \( r \) more negative, forcing the \( r \) estimate exponentially towards minus infinity, as shown Figure 53.

![Figure 53 – MIMNS pt. 1](image-url)
The physical interpretation of this result is interesting. The r-squared term originates in the $\omega \times (\omega \times r_{rel})$ cross product of Equation (2.1.5), which is the centripetal acceleration component of the overall relative acceleration equation. Carrying out the cross product yields a vector who's direction is opposite of $r_{rel}$, with sign always pointing towards the origin regardless of the direction of $\omega$. In this work, the accelerometers are positioned so that there is an $\overline{x}$ distance of one inch, meaning $a_{x,2}$ is one inch farther in the x-direction from the center of turning than $a_{x,1}$. Thus, the radial distance to $a_{x,2}$ is larger, and accordingly the centripetal acceleration caused by a rotation is larger, leading to a larger overall acceleration detected by the sensor.

If the $-r^2\overline{x}/\overline{x}$ is neglected, the result is that rotational acceleration $\dot{\omega}$ will always appear more positive than it truly is, due to the apparent imbalance in accelerations. Its inclusion, however, leads to the precarious feedback situation observed in Figure 53, although it should be noted that this affects all the MIMNS methods. The direct feedback method is particularly susceptible since the errors are integrated and then reintroduced, compounding the problem, and so it is recommended that this method not be used with the current configuration, and only with careful consideration in any future arrangements. One important observation is this effect can be minimized or eliminated if the placement of the accelerometers is selected such that $\overline{x}$ is as close to zero as possible. The maximization of $\overline{y}_x$, already shown to be extremely beneficial up to the point where rigidity is lost, also has a positive effect. Indeed this observation can be extended to the three dimensional equations, where careful placement of the accelerometers in each of
the three dimensions during design can drastically reduce or eliminate several terms from the kinematics and simplify the results.

Figure 53 also shows that the “via Velocity”, “gyro feedback”, and “Velocity Feedback” methods yielding slightly more stable and accurate results. Figure 54 examines a different portion of the same data, initialized during the static period occurring at 50-seconds into the trial. At around 54-seconds accelerations occur causing induced errors in the estimates that provide more insight into the estimation techniques. The direct feedback method performs as expected according to the discussion above, with the positive induced error slowly decaying towards zero over time. The “via Velocity” and gyro feedback methods produce similar results, with a large initial offset a very slow decay in the error. However the velocity feedback method shows a rapid divergence, due to the integration of the induced bias in the velocity method. The velocity estimation method, whether directly estimating rotation or feeding back into the rotational acceleration equation is particularly sensitive to changes in accelerometer biases due to the effect of the integrator.
While the introduction of large errors brought some useful observations that might otherwise go unnoticed over relatively short trial durations, further analysis of the effectiveness of the MIMNS theory or any of the particular techniques is prohibited by their presence. Time constraints prevent any further modifications or trials to the instruments, and so this section concludes with theories as to the source of the current anomalies.
First, it is believed that vibration of the chassis is causing undesirable effects. Further vibration characterization and isolation would likely benefit the process greatly, along with significant stiffening of the chassis, which not only would help to reduce transmission of low frequency vibrations but would also alleviate concerns about the effects of relative accelerations and velocities, which are contributing factors to the observed errors. Viscoelastic foams or other materials with “good” damping properties are worth further investigation.

Another potential source of error is the timing cycle on the duty cycle modulation (DCM) output of the accelerometers. The current circuit contains a 1.25-MΩ resistor for $R_{\text{set}}$, setting the period of the duty cycle output to nominally 10-ms; however, as much as 15-20 ms of delay from the time the output from one accelerometer is recorded to the recording of the next may occur, not accounting for delays caused by the microcontroller, which are probably not significant by comparison. If the accelerations are changing when this occurs, then an artificial component is created in the difference between any two acceleration measurements leading to a significant impact on the MIMNS theory. It may be possible to characterize these delays and compensate for them in the post-processing model; or, if the same type of accelerometer is used in the future (ADXL202’s from Analog Devices,) any of several modifications might be of use. The period of the duty cycle can further be reduced by an order of magnitude or more by varying the value of the $R_{\text{set}}$ resistor; or, rather than using the DCM output, utilize an ADC to read the analog output of the instrument directly. Details on performing these modifications can be found the datasheet for the instrument [25].
The passband on the output filter may also introduce error. Because of the low sampling rate capabilities of the basic stamp, a correspondingly low filter bandwidth must be used. Although an equation is provided for calculating the necessary capacitance for any passband, and no minimum bandpass frequency is specified in the datasheet, the lowest entry in an accompanying reference table is for a 10-Hz bandwidth filter utilizing a .47-μF capacitor, about double implemented this application. Using a bandwidth as low as 5-Hz was necessary to avoid aliasing; however, if the frequency is too low the filter will remove the frequencies of the signal that are being measured. Rather than attempting to lower the bandwidth any further, it may be beneficial to increase it, which would require an increase in sampling rate. In order to accomplish this, new computer architecture needs to be implemented, since the Basic Stamp currently performing data acquisition is sampling at its peak rate.

5.4.3 Optimization of Accelerometer Errors

One further avenue of experimentation was pursued, as described at the end of section 2.1. Once reasonable MIMNS data is being collected, the data can be compared to a calibrated gyro measurement for the same time period, and through the use of a computer optimization routine, errors in the accelerometer signals can be estimated. Reviewing the documentation for the ADXL202 accelerometers [25], two sources of calibration error are observed common to all XL202’s: a scale factor and a zero bias. The manual suggests a method for determining these scale factors that involves rotating the accelerometers 180° through the gravity vector, and comparing the outputs to the known value of ±g.
Figure 55 shows a Simulink block diagram used by MATLAB to process accelerometer recordings into a MIMNS estimate. Each accelerometer input (blocks ‘time_accel_1’ and ‘time_accel_2’) has a scale factor applied to it in the form of a signal gain (blocks ‘accel_1_sf’ and ‘accel_2_sf’), and a bias correction in the form of a constant that is subtracted from the signal (blocks ‘accel_1_bias’ and ‘accel_2_bias’). If these values were known in advance, by following the calibration method from [25], they could be inputted directly into this block diagram to improve the quality of the MIMNS measurement. In this case, since biases and scale factors are assumed to be unknown a priori, the values are left in the form of variables that can be manipulated by MATLAB.

An .m-file routine was created that varies the error-source variables using the "fminsearch" function (see section 2.1 for reference,) seeking the combination that will cause the MIMNS output to most closely match the gyro readings for a given segment of robot movement according to the cost function described in Equation (2.1.17).
Figure 55 - Accelerator Optimization Block Diagram
A simple test case was conducted where the robot was rotated approximately 90° in the counter-clockwise direction, followed by a short pause and a rotation in the clockwise direction back to the original heading. The error variables were initialized in the optimization assuming no error in the signal (i.e. a scale factor of one and a bias of zero,) and the output from the .m-file returned the optimized scale factors. Accelerometer 1 was returned with a scale factor of 0.77158 and a bias offset of -0.00039 ft/s², while Accelerometer 2 returned a scale factor of 0.87297 and a bias offset of 0.01168 ft/s². Figure 56 shows the MIMNS equations both before and after this optimization procedure, along with the gyro output for the same data. A significant improvement was observed using the biases and scale factors calculated by MIMNS comparison.
To further verify the validity of the new method, the calibration procedure from [25] was performed on the accelerometers, recording the readings as they are rotated through the gravity vector. This rough method, carried out by hand, yielded scale factors of 0.80 and 0.88 respectively for Accelerometers 1 and 2, close to the scale factors predicted by the MIMNS method that generated the plot above. Measurements taken did not have a tight enough tolerance to yield bias predictions of significance. These results however, are encouraging and with future refinements to the MIMNS data collection hardware a far more significant degree of accuracy (both in prediction and verification) might be accomplished.
6 Conclusions and Recommendations

An existing robotic platform was upgraded and repaired to make it suitable for the testing of the hypothesis that rotation can be estimated independent of rate gyros by comparing accelerometer signals. Experiments were carried out in a two dimensional environment in order to test a simplified version of the equations developed in this document.

1. It was discovered that work by previous student groups had not adequately accounted for the effects of vibration and signal processing, and once these problems were identified, design improvements were implemented to work around them.

2. A truncated version of the equations showed an encouraging correlation between the estimate derived from accelerometer inputs and the calculated derivative of the rate gyro measurements, taken to be a truth model.

3. The placement of accelerometers relative to one another has a marked influence on the performance, and was identified as a primary design consideration. Specifically:

   a. The distance component in the direction that is perpendicular to both the instrument sensing direction and vectoral direction of the rotation being estimated (e.g. the $\overline{y}_x$ direction in the equation for $\dot{r}$) should be maximized as much as possible while preserving rigidity.

   b. The other two direction components (e.g. $\overline{x}_x$ and $\overline{z}_x$ in the equation for $\dot{r}$) should be minimized.
4. A Kalman filter was investigated as a means of improving the estimates, but the necessarily limited nature of the system model prevented its implementation in a meaningful way.

5. An alternate version of the theory, based on velocities obtained from integrating the accelerometer signals before applying them to the MIMNS equations, was proposed and implemented for comparison.

6. Several modes of feedback to the acceleration-based version were also postulated and investigated. It was noted that if possible, these feedback terms should be minimized using geometry, as in conclusion 3-b.

7. The estimated rotation was used successfully to determine rough bias and scale factors for the accelerometer signals. These error factors were verified against standard calibration methods.

The quality of future work will depend strongly on making continued improvements to data acquisition. Recommendations for future work include:

- Find a new testing platform. Whether designed by a future researcher specifically for the purpose or possibly integrated into an existing vehicle from another vein of research, any further progress attempted on the current robotic platform will be limited.
- This researcher would strongly endorse the creation of an independent, stable, and reliable platform capable of carrying modular experiments as a project that would be of great benefit to future generations of RIT engineering students.
• A more powerful computer architecture, capable of sampling more data at faster rates, should be an important consideration. Integration into a real-time position estimation algorithm should be the eventual goal.

• The current choice in accelerometers, inherited from previous student groups, may not be appropriate for this application (see the “sticking” phenomena of section 5.4.1,) and a thorough investigation and trade study into various alternatives should be undertaken.

• A more functional system model, complete with deterministic control inputs, could be used to create a Kalman filter or other estimation tool to achieve more accurate real-time localization.

• The current design was created around the MIMNS cube device, but future designs should instead place the accelerometers in strategic locations in a chassis or frame according to the observations in conclusion 3 above.
7 Reference List

http://trueforce.com/Industrial_Manipulators_&_Automation/Industrial_Robotics _Companies/Adept_profile.htm


http://www.thetech.org/exhibits/online/robotics/universal/page06.html


http://www.doc.ic.ac.uk/~nd/surprise_95/journal/vol4/pma/report.html


http://ic.arc.nasa.gov/publications/pdf/smith90stochastic.pdf


Notes: Advisor: Dr. A. Crassidis

Notes: Advisor: Dr. A. Crassidis

Notes: Advisor: Dr. A. Crassidis


8 Appendices

8.1 Appendix A - Original Kalman Filter Simulation, MATLAB .m-file

format long
clear all
close all
clc

% ------------------ constants & variables ------------------

n = 1000;
T = .1;                % Sampling time (s)
r = [-8/12 -9/12; 7.5/12 -7.5/12; .5/12 -.5/12];
rel = r(:,2)-r(:,1);   % relative distance between accels
Xx = rel(1);
Yx = rel(2);
Zx = rel(3);

% ----------------- matrix definitions ---------------------

Ak = [1 0 0 ; 0 1 0 ; -T/Yx, T/Yx, 1];      % Discrete A matrix (initial ==> Ak(3,3) = 0)
H = [1 0 0 ; 0 1 0 ; 0 0 1];                % Measurement matrix
vk = [28.3*10^-6*32.2; 28.3*10^-6*32.2; 1.3798*10^-4];    % instrument noise vector
Rk(1,1) = cov(wgn(n,1,vk(1),'linear')) ;   % measurement noise matrix
Rk(2,2) = cov(wgn(n,1,vk(2),'linear'));
Rk(3,3) = cov(wgn(n,1,vk(3),'linear'));
phis = .01;
Qk = phis * [T 0 -(T^2)/(2*Yx); 0 T T^2/(2*Yx); -(T^2)/(2*Yx) T^2/(2*Yx)
2*T^3/(3*Yx^2)];
                    % Discrete process noise matrix
Pkml = [10000 0 0; 0 10000 0; 0 0 10000];         % state covariance initial matrix

% ---------------------- truth model ------------------------

axctin = zeros(n,1);
    axctin(i) = sin(i/250*2*pi)/2;
end
for i = 352:362
    axctin(i) = 1;
end
phidottruth = zeros(n,1);                     \% true rotational velocity
for i = 252:351
    phidottruth(i) = pi/2/10;
end
phitruth = zeros(n,1);                       \% array creation
phidotdottruth = zeros (n,1);
for i = 2:n
    phidotdottruth(i) = (phidottruth(i)-phidottruth(i-1))/T; \% rotational acceleration (truth)
    phitruth(i) = phitruth(i-1) + .5*T*(phidottruth(i)+phidottruth(i-1)); \% rotation angle
end
for i = 1:n                                   \% transform to instrument axis
    for j = 1:2
        aixtruth(i,j) = axctin(i) + phidottruth(i)^2*r(1,j) + phidotdottruth(i)*r(2,j);
    end
end
vxt = zeros(n,2);
sxt = zeros(n,2);
xpost = zeros(n,2);
ypost = zeros(n,2);
for i = 2:n
    vxt(i) = vxt(i-1) + .5 * T * (axctin(i) + axctin(i-1));
sxt(i) = sxt(i-1) + .5 * T * (vxt(i) + vxt(i-1));
xpost(i) = xpost(i-1) + (sxt(i) - sxt(i-1))*sin(phitruth(i));
ypost(i) = ypost(i-1) + (sxt(i) - sxt(i-1))*cos(phitruth(i));
end
% --------------------------- Measurements ----------------------------

aixmeasured = aixtruth + wgn(n,2,vk(1),'linear');\% Measurements
phidotmeasured = phidottruth + wgn(n,1,vk(3),'linear')+.1;

for i = 1:n
    if i == 1
        \% Riccati Eqns
        M = Ak*Pkml*Ak' + Qk;
        K = M*H'*inv(H*M*H'+Rk);
        P = (eye(3)-K*H)*M;
        \% Kalman update
        xkm1 = [0; 0; 0];\% estimated Initial conditions
        xdot(:,i) = Ak*xkm1;
for state propagation
        xsys(:,i) = zeros(3,1) + .5*T*(xdot(:,i) + 0);\% runge-kutta integration
    end
end
127
xmeas = [aixmeasured(i,1); aixmeasured(i,2); phidotmeasured(i)]; measurement vector

\[
xk = xsys(:,i) + K*(xmeas - H*xsys(:,i));
\]

% Kalman filter equations
% storage

\[
ax1k(i) = xk(1);
ax2k(i) = xk(2);
phidotk(i) = xk(3);
\]

else

\[
Ak(3,3) = 1 - 2*phidotk(i-1)*T*Xx/Yx;
\]
% non-linearity update

\[
M = Ak*P*Ak' + Qk;
\]
% Ricatti Eqns

\[
K = M*H'*inv(H*M*H'+ Rk);
\]
% Kalman update

\[
P = (eye(3)-K*H)*M;
\]
% storage

\[
xkm1 = xk;
\]

\[
Aki = [0 0 0; -1/Yx 1/Yx -phidotk(i-1)*Xx/Yx];
\]
% non-linear

version for integration

\[
xdot(:,i) = Aki*xkm1;
\]
% state propagation via

\[
xsys(:,i) = xsys(:,i-1) + .5*T*(xdot(:,i) + xdot(:,i-1));
\]
% 2nd order runge-kutta

\[
xmeas = [aixmeasured(i,1); aixmeasured(i,2); phidotmeasured(i)];
\]

\[
xk = xsys(:,i) + K*(xmeas - H*xsys(:,i));
\]

\[
ax1k(i) = xk(1);
ax1k(i) = xk(2);
phidotk(i) = xk(3);
\]
end

end

phimeasured = zeros(n,1); % calculated angular states

phidotdotmeasured = zeros(n,1);

for i = 2:n

\[
phidotdotmeasured(i) = (phidotmeasured(i) - phidotmeasured(i-1))/T;
\]

\[
phimeasured(i) = phimeasured(i-1) + .5 * T * (phidotmeasured(i) + phidotmeasured(i-1));
\]
end

actx_m = zeros(n,2); % transform measurements to globally oriented

for i = 2:n

center of turning axis

for j = 1:2

\[
actx_m(i,j) = aixmeasured(i,j) - phidotmeasured(i)^2*r(1,j) - phidotdotmeasured(i)*r(2,j);
\]
end
end

vctx = zeros(n,2);
sctx = zeros(n,2);
xposglobal = zeros(n,2);
yposglobal = zeros(n,2);

for i = 2:n
    for j = 1:2
        vctx(i,j) = vctx(i-1,j) + .5 * T * (actx_m(i,j) + actx_m(i-1,j));
        sctx(i,j) = sctx(i-1,j) + .5 * T * (vctx(i,j) + vctx(i-1,j));
        xposglobal(i,j) = xposglobal(i-1,j) + (sctx(i,j) - sctx(i-1,j))*sin(phimeasured(i));
        yposglobal(i,j) = yposglobal(i-1,j) + (sctx(i,j) - sctx(i-1,j))*cos(phimeasured(i));
    end
end

figure(1),
subplot(1,1,1),plot((1:n)/10,phidotk,(1:n)/10,phidotmeasured,(1:n)/10,phidottruth), grid,
legend('kalman','measured','truth model')
8.2 Appendix B – Modified Kalman Filter .m-file

format long
clear all
close all
clc

% ----------------- constants & variables ------------------

n = 1000;
T = .1;                % Sampling time (s)
r = [-8/12 -9/12; 7.5/12 -7.5/12; .5/12 -.5/12];
%r = [-8/12 -9/12; 2.5 -2.5; .5/12 -.5/12];
rrel = r(:,2)-r(:,1);
Xx = rrel(1);
Yx = rrel(2);
Zx = rrel(3);

% ----------------- matrix definitions ----------------------

Ak = [1 0 0 0; 0 1 0 0; -T/Yx, T/Yx, 1 0; 0 0 0 1];   % Discrete A matrix (initial =>
Ak(3,3) = 0)
H = [1 0 0 0; 0 1 0 0; 0 0 1 1; 0 0 0 1];          % Measurement matrix
vk = [28.3*10^-6*32.2; 28.3*10^-6*32.2; 1.3798*10^-4; 0];   % instrument noise vector
Rk(1,1) = cov(wgn(n,1,vk(1),'linear')) ;       % measurement noise matrix
Rk(2,2) = cov(wgn(n,1,vk(2),'linear'));
Rk(3,3) = cov(wgn(n,1,vk(3),'linear'));
Rk(4,4) = Rk(3,3);                             
phis = 1;
Qk = phis * [T 0 -(T^2)/(2*Yx) 0; 0 T T^2/(2*Yx) 0;... 
-(T^2)/(2*Yx) T^2/(2*Yx) 2*T^3/(3*Yx^2) 0; 0 0 0 T];   % Discrete
process noise matrix
Pkm1 = [10000 0 0 0; 0 10000 0 0; 0 0 10000 0; 0 0 0 10000];
% state covariance initialization matrix

% ----------------- truth model -----------------------------

axctin = zeros(n,1);                             % true linear acceleration (locally oriented)
for i = 1:250
    axctin(i) = sin(i/250*2*pi)/2;
end
for i = 352:362
    axctin(i) = 1;
end
phidottruth = zeros(n,1);
for i = 252:351
    phidottruth(i) = pi/2/10;
end
phitruth = zeros(n,1);
phidotdottruth = zeros(n,1);
for i = 2:n
    phidotdottruth(i) = (phidottruth(i)-phidottruth(i-1))/T;  
    phitruth(i) = phitruth(i-1) + .5*T*(phidottruth(i)+phidottruth(i-1));
end
for i = 1:n
    for j = 1:2
        aixtruth(i,j) = axctin(i) + phidottruth(i)*r(1,j) + phidotdottruth(i)*r(2,j);
    end
end
vxt = zeros(n,2);
sxt = zeros(n,2);
xpost = zeros(n,2);
ypost = zeros(n,2);
for i = 2:n
    vxt(i) = vxt(i-1) + .5 * T * (axctin(i) + axctin(i-1));
    sxt(i) = sxt(i-1) + .5 * T * (vxt(i) + vxt(i-1));
    xpost(i) = xpost(i-1) + (sxt(i) - sxt(i-1))*sin(phitruth(i));
    ypost(i) = ypost(i-1) + (sxt(i) - sxt(i-1))*cos(phitruth(i));
end

% ----------------------- Measurements --------------------------
aixmeasured = aixtruth + wgn(n,2,vk(1),'linear');
phidotmeasured = phidottruth + wgn(n,1,vk(3),'linear')+.1;
for i = 1:n
    if i == 1
        % Riccati Eqns
        M = Ak*Pkml*Ak' + Qk;
        K = M*H'*inv(H*M*H' + Rk);
        P = (eye(4)-K*H)*M;
        % Kalman update
        xkm1 = [0; 0; 0; .2];
        % estimated Initialization conditions
        xdot(:,i) = Ak*xkm1;
        % runge-kutta integration for state propagation
        xsys(:,i) = zeros(4,1) + .5*T*(xdot(:,i) + 0);
        xmeas = [aixmeasured(i,1); aixmeasured(i,2); phidotmeasured(i); xkm1(4)];
% measurement vector
xk = xsys(:,i) + K*(xmeas - H*xsys(:,i));
% Kalman filter equations
ax1k(i) = xk(1);
ax2k(i) = xk(2);
phidotk(i) = xk(3);
bias(i) = xk(4);
% storage
else
% non-linearity update
Ak(3,3) = 1 - 2*phidotk(i-1)*T*Xx/Yx;
%M = Ak*P*Ak' + Qk;
K = M*H'inv(H*M*H' + Rk);
P = (eye(4)-K*H)*M;
% Kalman update
xkml = xk;
Aki = [0 0 0 0; 0 0 0 0; -1/Yx 1/Yx -phidotk(i-1)*Xx/Yx 0; 0 0 0];
% non-linear version for integration
xdot(:,i) = Aki*xkml;
% state propagation via
xsys(:,i) = xsys(:,i-1) + .5*T*(xdot(:,i) + xdot(:,i-1));
% 2nd order runge-kutta
xmeas = [aixmeasured(i,1); aixmeasured(i,2); ...
        phidotmeasured(i); phidotmeasured(i-1)-phidotk(i-1)];
xk = xsys(:,i) + K*(xmeas - H*xsys(:,i));
ax1k(i) = xk(1);
ax2k(i) = xk(2);
phidotk(i) = xk(3);
bias(i) = xk(4);
end
end

phimeasured = zeros(n,1); % calculated angular states
phidotdotmeasured = zeros(n,1);
for i = 2:n
    phidotdotmeasured(i) = (phidotmeasured(i) - phidotmeasured(i-1))/T;
    phimeasured(i) = phimeasured(i-1) + .5 * T * (phidotmeasured(i) + phidotmeasured(i-1));
end

actx_m = zeros(n,2); % transform to center of turning axis
for i = 2:n
    for j = 1:2
        actx_m(i,j) = aixmeasured(i,j) - phidotmeasured(i)^2*r(1,j) -
                      phidotdotmeasured(i)*r(2,j);
    end
end

vctx = zeros(n,2);
sctx = zeros(n,2);
xposglobal = zeros(n,2);
yposglobal = zeros(n,2);

for i = 2:n
    for j = 1:2
        vctx(i,j) = vctx(i-1,j) + .5 * T * (actx_m(i,j) + actx_m(i-1,j));
        sctx(i,j) = sctx(i-1,j) + .5 * T * (vctx(i,j) + vctx(i-1,j));
        xposglobal(i,j) = xposglobal(i-1,j) + (sctx(i,j) - sctx(i-1,j))*sin(phimeasured(i));
        yposglobal(i,j) = yposglobal(i-1,j) + (sctx(i,j) - sctx(i-1,j))*cos(phimeasured(i));
    end
end

figure(1),
subplot(1,1,1),plot((1:n)/10,phidotk,(1:n)/10,phidotmeasured,(1:n)/10,phidottruth), grid,
legend('kalman','measured','truth model'), title('Rotation Rate r')
8.3 Appendix C – MIMNS .m-file

format long
clear all
close all
clc

% ----------------- constants & variables ---------------------

data = dlmread('data/data_unmarked.txt',','');
n = length(data);
T = .1;
% Sampling time (s)
r = [-8/12 -9/12; 7.5/12 -7.5/12];
rel = r(:,2)-r(:,1);
% relative distance between accels
Xx = rel(1);
Yx = rel(2);

% ------------------ Measurements --------------------------

phik = zeros(n,1);
% calculated angular states
phidotdotk = zeros(n,1);
actx = zeros(n,2);
vctx = zeros(n,2);
sctx = zeros(n,2);
xposk = zeros(n,2);
yposk = zeros(n,2);

i = 1;
k = 1;
x = 0;

while i <= n
    if data(i,1) == 0
        % row of zeros is signal of periodic stop
        i = i + 1;
        T2(1) = .287/data(i,2)/(.8*10^-6);
        % Accel calibration data
        T2(2) = .287/data(i,3)/(.8*10^-6);
        i = i + 1;
        a_zero(1) = 0;
        % calculate zero offsets
        a_zero(2) = 0;
        g_zero = 0;
        for j = i:(i+19)
            a_zero(1) = a_zero(1) + data(j,2);
            a_zero(2) = a_zero(2) + data(j,3);
            
    

134
g_zero = g_zero + data(j,1);
end
a_zero(1) = a_zero(1)/20;
a_zero(2) = a_zero(2)/20;
g_zero = g_zero/20;
x_zero(k,1) = .5 - a_zero(1)/T2(1);
x_zero(k,2) = .5 - a_zero(2)/T2(2);
x_zero(k,3) = .5 - g_zero/4096;
xmeas(k,1) = ((data(i,2) / T2(1) + x_zero(k,1) - .5) / .125)*32.2; % measurement vector
xmeas(k,2) = ((data(i,3) / T2(2) + x_zero(k,2) - .5) / .125)*32.2;
xmeas(k,3) = ((data(i,1)/4096 + x_zero(k,3) - .5)*5)/(.0125*180/pi);
i = i+1;
k = k+1;
else
x_zero(k,1) = x_zero(k-1,1);
x_zero(k,2) = x_zero(k-1,2);
x_zero(k,3) = x_zero(k-1,3);
xmeas(k,1) = ((data(i,2)/T2(1) + x_zero(k,1) - .5)/.125)*32.2; % measurement vector
xmeas(k,2) = ((data(i,3)/T2(2) + x_zero(k,2) - .5)/.125)*32.2;
xmeas(k,3) = ((data(i,1)/4096 + x_zero(k,3) - .5)*5)/(.0125*180/pi);
i = i+1;
k = k+1;
end
end
accel_1_bias = x(1);
accel_2_bias = x(1);
time = ((1:(length(xmeas)))/10)';
accel_1 = xmeas(:,1);
accel_2 = xmeas(:,2);
rate_gyro = xmeas(:,3);
start = 1;
stop = 400;
time = time(start:stop)-time(start);
accel_1 = accel_1(start:stop);
accel_2 = accel_2(start:stop);
rate_gyro = rate_gyro(start:stop);
sim('ag_sim')

figure(1)

 subplot(1,1,1),plot(time_sim,rate_gyro(1:length(time_sim)),time_sim,mimnsa,time_sim,mimnsv,time_sim,...

135
mimnsvf,time_sim,gyro), grid, legend('Raw gyro','MIMNS Direct Feedback','MIMNS via Velocity',...
'MIMNS Velocity Feedback','Gyro Feedback'),title('MIMNS'), xlabel('time (s)'),
ylabel('rad/s')

clear time accel_1 accel_2 rate_gyro
time=((1:(length(xmeas)))/10);
accel_1=xmeas(:,1);
accel_2=xmeas(:,2);
rate_gyro=xmeas(:,3);
start = 500
stop = 750
time=time(start:stop)-time(start);
accel_1=accel_1(start:stop);
accel_2=accel_2(start:stop);
rate_gyro=rate_gyro(start:stop);
sim('agsim')
figure(2)

subplot(1,1,1),plot(time_sim+50,rate_gyro(1:length(time_sim)),time_sim+50,mimnnsa_tim e_sim+50,mimnsv,time_sim+50,....
mimnsvf,time_sim+50,gyro), grid, legend('Raw gyro','MIMNS Direct Feedback','MIMNS via Velocity',...
'MIMNS Velocity Feedback','Gyro Feedback'), title('MIMNS'), xlabel('time (s)'),
ylabel('rad/s')
8.4 Simulink Diagram for r-feedback methods

Figure 57 - MIMNS Simulink Block Diagram
8.5 Appendix D – ADXL202 and ADXRS150 Datasheets
FEATURES
2-Axis Acceleration Sensor on a Single IC Chip
5 mm × 5 mm × 2 mm Ultrasmall Chip Scale Package
2 mg Resolution at 60 Hz
Low-Power < 0.6 mA
Direct Interface to Low-Cost Microcontrollers via Duty Cycle Output
BW Adjustment with a Single Capacitor
3 V to 5.25 V Single Supply Operation
1000 g Shock Survival
APPLICATIONS
2-Axis Tilt Sensing with Faster Response than Electrolytic, Mercury, or Thermal Sensors
Computer Peripherals
Information Appliances
Alarms and Motion Detectors
Disk Drives
Vehicle Security

GENERAL DESCRIPTION
The ADXL202E is a low-cost, low-power, complete 2-axis accelerometer with a digital output, all on a single monolithic IC. It is an improved version of the ADXL202AQC/JQC. The ADXL202E will measure accelerations with a full-scale range of ±2 g. The ADXL202E can measure both dynamic acceleration (e.g., vibration) and static acceleration (e.g., gravity).

The outputs are analog voltage or digital signals whose duty cycles (ratio of pulselwidth to period) are proportional to acceleration. The duty cycle outputs can be directly measured by a microprocessor counter, without an A/D converter or glue logic. The duty cycle period is adjustable from 0.5 ms to 10 ms via a single resistor (RSET).

The typical noise floor is 200 μg/√Hz, allowing signals below 2 mg (at 60 Hz bandwidth) to be resolved.

The bandwidth of the accelerometer is set with capacitors CX and CY at the X_FILT and Y_FILT pins. An analog output can be reconstructed by filtering the duty cycle output.

The ADXL202E is available in 5 mm × 5 mm × 2 mm 8-lead hermetic LCC package.

*Patents Pending

REV. A

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.
## ADXL202E—SPECIFICATIONS

(T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>, T<sub>A</sub> = 25°C for J Grade only, V<sub>DD</sub> = 5 V, R<sub>SET</sub> = 125 kΩ, Acceleration = 0 g, unless otherwise noted.)

### Parameters and Conditions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Conditions</th>
<th>TPC&lt;sup&gt;1&lt;/sup&gt; Graph</th>
<th>ADXL202E</th>
<th>ADXL202AE</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SENSOR INPUT</strong></td>
<td></td>
<td></td>
<td>Min</td>
<td>Typ</td>
<td>Max</td>
</tr>
<tr>
<td>Measurement Range&lt;sup&gt;2&lt;/sup&gt;</td>
<td>Each Axis</td>
<td>±2</td>
<td>±2</td>
<td>±2</td>
<td>g</td>
</tr>
<tr>
<td>Nonlinearity</td>
<td>Best Fit Straight Line</td>
<td>X</td>
<td>0.2</td>
<td>0.2</td>
<td>% of FS</td>
</tr>
<tr>
<td>Alignment Error&lt;sup&gt;3&lt;/sup&gt;</td>
<td>X Sensor to Y Sensor</td>
<td>X</td>
<td>±1</td>
<td>±1</td>
<td>Degrees</td>
</tr>
<tr>
<td>Cross-Axis Sensitivity&lt;sup&gt;4&lt;/sup&gt;</td>
<td></td>
<td></td>
<td>0.01</td>
<td>0.01</td>
<td>Degrees</td>
</tr>
<tr>
<td><strong>SENSITIVITY</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duty Cycle per g&lt;sup&gt;g&lt;/sup&gt;</td>
<td>Each Axis</td>
<td>X</td>
<td>10.5</td>
<td>12.5</td>
<td>14.5</td>
</tr>
<tr>
<td>Duty Cycle per g&lt;sup&gt;g&lt;/sup&gt;</td>
<td>T1/T2, V&lt;sub&gt;DD&lt;/sub&gt; = 5 V</td>
<td>X</td>
<td>10</td>
<td>12.5</td>
<td>15</td>
</tr>
<tr>
<td>Sensitivity X&lt;sub&gt;FILT&lt;/sub&gt;, Y&lt;sub&gt;FILT&lt;/sub&gt;</td>
<td>T1/T2, V&lt;sub&gt;DD&lt;/sub&gt; = 3 V</td>
<td>X</td>
<td>8.5</td>
<td>11</td>
<td>13.5</td>
</tr>
<tr>
<td>Sensitivity X&lt;sub&gt;FILT&lt;/sub&gt;, Y&lt;sub&gt;FILT&lt;/sub&gt;</td>
<td>V&lt;sub&gt;DD&lt;/sub&gt; = 5 V</td>
<td>X</td>
<td>250</td>
<td>312</td>
<td>375</td>
</tr>
<tr>
<td>Temperture Drift&lt;sup&gt;5&lt;/sup&gt;</td>
<td>Delta from 25°C</td>
<td>X</td>
<td>140</td>
<td>167</td>
<td>195</td>
</tr>
<tr>
<td><strong>ZERO g BIAS LEVEL</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 g Duty Cycle</td>
<td>Each Axis</td>
<td>X</td>
<td>34</td>
<td>50</td>
<td>66</td>
</tr>
<tr>
<td>0 g Duty Cycle</td>
<td>T1/T2, V&lt;sub&gt;DD&lt;/sub&gt; = 5 V</td>
<td>X</td>
<td>31</td>
<td>50</td>
<td>69</td>
</tr>
<tr>
<td>0 g Voltage X&lt;sub&gt;FILT&lt;/sub&gt;, Y&lt;sub&gt;FILT&lt;/sub&gt;</td>
<td>T1/T2, V&lt;sub&gt;DD&lt;/sub&gt; = 3 V</td>
<td>X</td>
<td>2.1</td>
<td>2.5</td>
<td>2.9</td>
</tr>
<tr>
<td>0 g Voltage X&lt;sub&gt;FILT&lt;/sub&gt;, Y&lt;sub&gt;FILT&lt;/sub&gt;</td>
<td>V&lt;sub&gt;DD&lt;/sub&gt; = 5 V</td>
<td>X</td>
<td>1.2</td>
<td>1.5</td>
<td>1.8</td>
</tr>
<tr>
<td>0 g Duty Cycle vs. Supply</td>
<td>Delta from 25°C</td>
<td>X</td>
<td>1.0</td>
<td>4.0</td>
<td>0</td>
</tr>
<tr>
<td>0 g Offset vs. Temperature&lt;sup&gt;6&lt;/sup&gt;</td>
<td>Delta from 25°C</td>
<td>X</td>
<td>2.0</td>
<td>2.0</td>
<td>0</td>
</tr>
<tr>
<td><strong>NOISE PERFORMANCE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noise Density</td>
<td>@ 25°C</td>
<td>X</td>
<td>200</td>
<td>200</td>
<td>1000</td>
</tr>
<tr>
<td><strong>FREQUENCY RESPONSE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 dB Bandwidth</td>
<td>At Pins X&lt;sub&gt;FILT&lt;/sub&gt;, Y&lt;sub&gt;FILT&lt;/sub&gt;</td>
<td>6</td>
<td>6</td>
<td>10</td>
<td>kHz</td>
</tr>
<tr>
<td>Sensor Resonant Frequency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td><strong>FILTER</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R&lt;sub&gt;FILT&lt;/sub&gt; Tolerance</td>
<td>32 kΩ Nominal</td>
<td>±15</td>
<td>±15</td>
<td>±15</td>
<td>%</td>
</tr>
<tr>
<td>Minimum Capacitance</td>
<td>At Pins X&lt;sub&gt;FILT&lt;/sub&gt;, Y&lt;sub&gt;FILT&lt;/sub&gt;</td>
<td>1000</td>
<td>1000</td>
<td></td>
<td>pF</td>
</tr>
<tr>
<td><strong>SELF-TEST</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duty Cycle Change</td>
<td>Self-Test &quot;0&quot; to &quot;1&quot;</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>%</td>
</tr>
<tr>
<td><strong>DUTY CYCLE OUTPUT STAGE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F&lt;sub&gt;SET&lt;/sub&gt; = 125 kΩ</td>
<td>R&lt;sub&gt;SET&lt;/sub&gt; = 125 kΩ</td>
<td>0.7</td>
<td>1.3</td>
<td>0.7</td>
<td>1.3</td>
</tr>
<tr>
<td>Output High Voltage</td>
<td>I = 25 µA</td>
<td>0.5 – 200 mV</td>
<td>0.5</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>Output Low Voltage</td>
<td>I = 25 µA</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>T2 Drift vs. Temperature</td>
<td>Rise/Fall Time</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td><strong>POWER SUPPLY</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating Voltage Range</td>
<td>C&lt;sub&gt;FILT&lt;/sub&gt; in µF</td>
<td>3</td>
<td>5.25</td>
<td>3.0</td>
<td>5.25</td>
</tr>
<tr>
<td>Quiescent Supply Current</td>
<td></td>
<td>0.6</td>
<td>1.0</td>
<td>0.6</td>
<td>1.0</td>
</tr>
<tr>
<td>Turn-On Time</td>
<td>C&lt;sub&gt;FILT&lt;/sub&gt; in µF</td>
<td>160 × C&lt;sub&gt;FILT&lt;/sub&gt; + 0.3</td>
<td>160 × C&lt;sub&gt;FILT&lt;/sub&gt; + 0.3</td>
<td>ms</td>
<td></td>
</tr>
<tr>
<td><strong>TEMPERATURE RANGE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specified Performance AE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating Range</td>
<td></td>
<td>0</td>
<td>-40</td>
<td>+85</td>
<td>°C</td>
</tr>
</tbody>
</table>

**Notes**

1. Typical Performance Characteristics.
2. Guaranteed by measurement of initial offset and sensitivity.
3. Alignment error is specified as the angle between the true and indicated axis of sensitivity (see TPC 15).
4. Cross-axis sensitivity is the algebraic sum of the alignment and the inherent sensitivity errors.
5. Defined as the output change from ambient to maximum temperature or ambient to minimum temperature.

Specifications subject to change without notice.
**ABSOLUTE MAXIMUM RATINGS**

**Acceleration (Any Axis, Unpowered for 0.5 ms):** 1000 g

**Acceleration (Any Axis, Powered for 0.5 ms):** 500 g

+V5 = -0.3 V to +6.0 V

Output Short Circuit Duration, (Any Pin to Common) = Indefinite

Operating Temperature = -55°C to +125°C

Storage Temperature = -65°C to +150°C

*Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Drops onto hard surfaces can cause shocks of greater than 1000 g and exceed the absolute maximum rating of the device. Care should be exercised in handling to avoid damage.

**Package Characteristics**

<table>
<thead>
<tr>
<th>Package</th>
<th>Weight</th>
<th>θJA</th>
<th>θJC</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-Lead LCC</td>
<td>120°C/W</td>
<td>tbd°C/W</td>
<td>&lt;1.0 grams</td>
<td></td>
</tr>
</tbody>
</table>

**PIN FUNCTION DESCRIPTIONS**

<table>
<thead>
<tr>
<th>Pin No.</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ST</td>
<td>Self-Test</td>
</tr>
<tr>
<td>2</td>
<td>T2</td>
<td>Connect RSET to Set T2 Period</td>
</tr>
<tr>
<td>3</td>
<td>COM</td>
<td>Common</td>
</tr>
<tr>
<td>4</td>
<td>YOUT</td>
<td>Y-Channel Duty Cycle Output</td>
</tr>
<tr>
<td>5</td>
<td>XOUT</td>
<td>X-Channel Duty Cycle Output</td>
</tr>
<tr>
<td>6</td>
<td>YFILT</td>
<td>Y-Channel Filter Pin</td>
</tr>
<tr>
<td>7</td>
<td>XFILT</td>
<td>X-Channel Filter Pin</td>
</tr>
<tr>
<td>8</td>
<td>VDD</td>
<td>3 V to 5.25 V</td>
</tr>
</tbody>
</table>

**ORDERING GUIDE**

<table>
<thead>
<tr>
<th>Model</th>
<th>No. of Axes</th>
<th>Specified Voltage</th>
<th>Temperature Range</th>
<th>Package Description</th>
<th>Package Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADXL202JE</td>
<td>2</td>
<td>3 V to 5 V</td>
<td>0 to 70°C</td>
<td>8-Lead LCC</td>
<td>E-8</td>
</tr>
<tr>
<td>ADXL202AE</td>
<td>2</td>
<td>3 V to 5 V</td>
<td>-40°C to +85°C</td>
<td>8-Lead LCC</td>
<td>E-8</td>
</tr>
</tbody>
</table>

**CAUTION**

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the ADXL202E features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high-energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.
ADXL202E—Typical Performance Characteristics*

*Data taken from 4500 parts over 3 lots minimum.
\[ V_{DD} = 3 \text{ V} \]

TPC 7. Y-Axis Sensitivity Distribution at \( Y_{\text{FILT}} \), \( V_{DD} = 3 \text{ V} \)

\[ V_{DD} = 5 \text{ V} \]

TPC 10. Y-Axis Sensitivity Distribution at \( Y_{\text{FILT}} \), \( V_{DD} = 5 \text{ V} \)

TPC 8. X-Axis Sensitivity at \( X_{\text{OUT}} \), \( V_{DD} = 3 \text{ V} \)

TPC 11. X-Axis Sensitivity at \( X_{\text{OUT}} \), \( V_{DD} = 5 \text{ V} \)

TPC 9. Y-Axis Sensitivity at \( Y_{\text{OUT}} \), \( V_{DD} = 3 \text{ V} \)

TPC 12. Y-Axis Sensitivity at \( Y_{\text{OUT}} \), \( V_{DD} = 5 \text{ V} \)
TPC 13. Noise Density Distribution, $V_{DD} = 3$ V

TPC 16. Noise Density Distribution, $V_{DD} = 5$ V

TPC 14. Typical Supply Current vs. Temperature

TPC 17. Cross-Axis Sensitivity Distribution

TPC 15. Rotational Die Alignment

TPC 18. Typical Turn-On Time
TPC 19. X-Axis Zero g Drift Due to Temperature Distribution, –40°C to +85°C

TPC 20. X-Axis Sensitivity Drift at $X_{FILT}$ Due to Temperature Distribution, –40°C to +85°C

TPC 21. Typical X-Axis Zero g vs. Output for 16 Parts

TPC 22. Y-Axis Zero g Drift Due to Temperature Distribution, –40°C to +85°C

TPC 23. Y-Axis Sensitivity Drift at $Y_{FILT}$ Due to Temperature Distribution, –40°C to +85°C

TPC 24. Typical Y-Axis Zero g vs. Output for 16 Parts
ADXL202E

DEFINITIONS

T1  Length of the “on” portion of the cycle.
T2  Length of the total cycle.
Duty Cycle  Ratio of the “on” time (T1) of the cycle to the total cycle (T2). Defined as T1/T2 for the ADXL202E/ADXL210.
Pulsewidth  Time period of the “on” pulse. Defined as T1 for the ADXL202E/ADXL210.

THEROY OF OPERATION

The ADXL202E is a complete, dual-axis acceleration measurement system on a single monolithic IC. It contains a polysilicon surface-micromachined sensor and signal conditioning circuitry to implement an open loop acceleration measurement architecture. For each axis, an output circuit converts the analog signal to a duty cycle modulated (DCM) digital signal that can be decoded with a counter/timer port on a microprocessor. The ADXL202E is capable of measuring both positive and negative accelerations to at least ±2 g. The accelerometer can measure static acceleration forces such as gravity, allowing it to be used as a tilt sensor.

The sensor is a surface micromachined polysilicon structure built on top of the silicon wafer. Polysilicon springs suspend the structure over the surface of the wafer and provide a resistance against acceleration forces. Deflection of the structure is measured using a differential capacitor that consists of independent fixed plates and central plates attached to the moving mass. The fixed plates are driven by 180° out of phase square waves. An acceleration will deflect the beam and unbalance the differential capacitor, resulting in an output square wave whose amplitude is proportional to acceleration. Phase sensitive demodulation techniques are then used to rectify the signal and determine the direction of the acceleration.

The output of the demodulator drives a duty cycle modulator (DCM) stage through a 32 kΩ resistor. At this point a pin is available on each channel to allow the user to set the signal bandwidth of the device by adding a capacitor. This filtering improves measurement resolution and helps prevent aliasing.

After being low-pass filtered, the analog signal is converted to a duty cycle modulated signal by the DCM stage. A single resistor sets the period for a complete cycle (T2), which can be set between 0.5 ms and 10 ms (see Figure 12). A 0 g acceleration produces a nominally 50% duty cycle. The acceleration signal can be determined by measuring the length of the T1 and T2 pulses with a counter/timer or with a polling loop using a low cost microcontroller.

An analog output voltage can be obtained either by buffering the signal from the X\text{FILT} and Y\text{FILT} pin, or by passing the duty cycle signal through an RC filter to reconstruct the dc value. The ADXL202E will operate with supply voltages as low as 3.0 V or as high as 5.25 V.

APPLICATIONS

POWER SUPPLY DECOUPLING

For most applications a single 0.1 μF capacitor, C\text{DC}, will adequately decouple the accelerometer from signal and noise on the power supply. However, in some cases, especially where digital devices such as microcontrollers share the same power supply, digital noise on the supply may cause interference on the ADXL202E output. This may be observed as a slowly undulating fluctuation of voltage at X\text{FILT} and Y\text{FILT}. If additional decoupling is needed, a 100 Ω (or smaller) resistor or ferrite beads, may be inserted in the supply line of the ADXL202E.

Figure 1. Typical Output Duty Cycle

Figure 2.
DESIGN PROCEDURE FOR THE ADXL202E

The design procedure for using the ADXL202E with a duty cycle output involves selecting a duty cycle period and a filter capacitor. A proper design will take into account the application requirements for bandwidth, signal resolution and acquisition time, as discussed in the following sections.

Decoupling Capacitor $C_{DC}$

A 0.1 μF capacitor is recommended from $V_{DD}$ to COM for power supply decoupling.

ST

The ST pin controls the self-test feature. When this pin is set to $V_{DD}$, an electrostatic force is exerted on the beam of the accelerometer. The resulting movement of the beam allows the user to test if the accelerometer is functional. The typical change in output will be 10% at the duty cycle outputs (corresponding to 800 mg). This pin may be left open circuit or connected to common in normal use.

Duty Cycle Decoding

The ADXL202E's digital output is a duty cycle modulator. Acceleration is proportional to the ratio $T_1/T_2$. The nominal output of the ADXL202E is:

$$0 \ g = 50\% \ \text{Duty Cycle}$$

Scale factor is 12.5% Duty Cycle Change per $g$

These nominal values are affected by the initial tolerance of the device including zero $g$ offset error and sensitivity error.

$T_2$ does not have to be measured for every measurement cycle. It need only be updated to account for changes due to temperature, (a relatively slow process). Since the $T_2$ time period is shared by both X and Y channels, it is necessary only to measure it on one channel of the ADXL202E. Decoding algorithms for various microcontrollers have been developed. Consult the appropriate Application Note.

Setting the Bandwidth Using $C_X$ and $C_Y$

The ADXL202E has provisions for bandlimiting the $X_{FILT}$ and $Y_{FILT}$ pins. Capacitors must be added at these pins to implement low-pass filtering for antialiasing and noise reduction. The equation for the 3 dB bandwidth is:

$$F_{-3dB} = \frac{1}{(2 \pi (32 \ \Omega) \times C(x, y))}$$

or, more simply, $F_{-3dB} = \frac{5 \mu F}{C(x, y)}$

The tolerance of the internal resistor ($R_{FILT}$), can vary typically as much as ±15% of its nominal value of 32 kΩ; so the bandwidth will vary accordingly. A minimum capacitance of 1000 pF for $C(x, y)$ is required in all cases.

### Table I. Filter Capacitor Selection, $C_X$ and $C_Y$

<table>
<thead>
<tr>
<th>Bandwidth</th>
<th>Capacitor Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Hz</td>
<td>0.47 μF</td>
</tr>
<tr>
<td>50 Hz</td>
<td>0.10 μF</td>
</tr>
<tr>
<td>100 Hz</td>
<td>0.05 μF</td>
</tr>
<tr>
<td>200 Hz</td>
<td>0.027 μF</td>
</tr>
<tr>
<td>500 Hz</td>
<td>0.01 μF</td>
</tr>
<tr>
<td>5 kHz</td>
<td>0.001 μF</td>
</tr>
</tbody>
</table>

Setting the DCM Period with $R_{SET}$

The period of the DCM output is set for both channels by a single resistor from $R_{SET}$ to ground. The equation for the period is:

$$T_2 = \frac{R_{SET}}{125 \ \text{MΩ}}$$

A 125 kΩ resistor will set the duty cycle repetition rate to approximately 1 kHz, or 1 ms. The device is designed to operate at duty cycle periods between 0.5 ms and 10 ms.

### Table II. Resistor Values to Set $T_2$

<table>
<thead>
<tr>
<th>$T_2$</th>
<th>$R_{SET}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ms</td>
<td>125 kΩ</td>
</tr>
<tr>
<td>2 ms</td>
<td>250 kΩ</td>
</tr>
<tr>
<td>5 ms</td>
<td>625 kΩ</td>
</tr>
<tr>
<td>10 ms</td>
<td>1.25 MΩ</td>
</tr>
</tbody>
</table>

Note that the $R_{SET}$ should always be included, even if only an analog output is desired. Use an $R_{SET}$ value between 500 kΩ and 2 MΩ when taking the output from $X_{FILT}$ or $Y_{FILT}$. The $R_{SET}$ resistor should be placed close to the $T_2$ pin to minimize parasitic capacitance at this node.

Selecting the Right Accelerometer

For most tilt sensing applications the ADXL202E is the most appropriate accelerometer. Its higher sensitivity (12.5%/g) allows the user to use a lower speed counter for PWM decoding while maintaining high resolution. The ADXL210 should be used in applications where accelerations of greater than ±2 g are expected.
ADXL202E

MICROCOMPUTER INTERFACES

The ADXL202E is specifically designed to work with low-cost microcontrollers. Specific code sets, reference designs, and application notes are available from the factory. This section will outline a general design procedure and discuss the various trade-offs that need to be considered.

The designer should have some idea of the required performance of the system in terms of:

Resolution: the smallest signal change that needs to be detected.

Bandwidth: the highest frequency that needs to be detected.

Acquisition Time: the time that will be available to acquire the signal on each axis.

These requirements will help to determine the accelerometer bandwidth, the speed of the microcontroller clock and the length of the T2 period.

When selecting a microcontroller it is helpful to have a counter timer port available. The microcontroller should have provisions for software calibration. While the ADXL202E is a highly accurate accelerometer, it has a wide tolerance for initial offset. The easiest way to null this offset is with a calibration factor saved on the microcontroller or by a user calibration for zero g. In the case where the offset is calibrated during manufacture, there are several options, including external EEPROM and microcontrollers with "one-time programmable" features.

DESIGN TRADE-OFFS FOR SELECTING FILTER CHARACTERISTICS: THE NOISE/BW TRADE-OFF

The accelerometer bandwidth selected will determine the measurement resolution (smallest detectable acceleration). Filtering can be used to lower the noise floor and improve the resolution of the accelerometer. Resolution is dependent on both the analog filter bandwidth at Xfilt and Yfilt and on the speed of the microcontroller counter.

The analog output of the ADXL202E has a typical bandwidth of 5 kHz, while the duty cycle modulators' bandwidth is 500 Hz. The user must filter the signal at this point to limit aliasing errors. To minimize DCM errors the analog bandwidth should be less than 1/10 the DCM frequency. Analog bandwidth may be increased to up to 1/2 the DCM frequency in many applications. This will result in greater dynamic error generated at the DCM.

The analog bandwidth may be further decreased to reduce noise and improve resolution. The ADXL202E noise has the characteristics of white Gaussian noise that contributes equally at all frequencies and is described in terms of μg per root Hz; i.e., the noise is proportional to the square root of the bandwidth of the accelerometer. It is recommended that the user limit bandwidth to the lowest frequency needed by the application, to maximize the resolution and dynamic range of the accelerometer.

With the single pole roll-off characteristic, the typical noise of the ADXL202E is determined by the following equation:

\[ \text{Noise (rms)} = \left(200 \ \mu g/\sqrt{\text{Hz}}\right) \times \left(\sqrt{\text{BW} \times 1.6}\right) \]

At 100 Hz the noise will be:

\[ \text{Noise (rms)} = \left(200 \ \mu g/\sqrt{\text{Hz}}\right) \times \left(\sqrt{100 \times (1.6)}\right) = 2.53 \text{ mg} \]

Often the peak value of the noise is desired. Peak-to-peak noise can only be estimated by statistical methods. Table III is useful for estimating the probabilities of exceeding various peak values, given the rms value.

<table>
<thead>
<tr>
<th>Nominal Peak-to-Peak Value</th>
<th>% of Time that Noise Will Exceed Nominal Peak-to-Peak Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0 × rms</td>
<td>32%</td>
</tr>
<tr>
<td>4.0 × rms</td>
<td>4.6%</td>
</tr>
<tr>
<td>6.0 × rms</td>
<td>0.27%</td>
</tr>
<tr>
<td>8.0 × rms</td>
<td>0.006%</td>
</tr>
</tbody>
</table>

The peak-to-peak noise value will give the best estimate of the uncertainty in a single measurement.

Table IV gives typical noise output of the ADXL202E for various CX and CY values.

<table>
<thead>
<tr>
<th>Bandwidth</th>
<th>CX, CY</th>
<th>rms Noise</th>
<th>Peak-to-Peak Noise Estimate 95% Probability (rms × 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Hz</td>
<td>0.47 μF</td>
<td>0.8 mg</td>
<td>3.2 mg</td>
</tr>
<tr>
<td>50 Hz</td>
<td>0.10 μF</td>
<td>1.8 mg</td>
<td>7.2 mg</td>
</tr>
<tr>
<td>100 Hz</td>
<td>0.05 μF</td>
<td>2.5 mg</td>
<td>10.1 mg</td>
</tr>
<tr>
<td>200 Hz</td>
<td>0.027 μF</td>
<td>3.6 mg</td>
<td>14.3 mg</td>
</tr>
<tr>
<td>500 Hz</td>
<td>0.01 μF</td>
<td>5.7 mg</td>
<td>22.6 mg</td>
</tr>
</tbody>
</table>

CHOOSING T2 AND COUNTER FREQUENCY: DESIGN TRADE-OFFS

The noise level is one determinant of accelerometer resolution. The second relates to the measurement resolution of the counter when decoding the duty cycle output.

The ADXL202E's duty cycle converter has a resolution of approximately 14 bits; better resolution than the accelerometer itself. The actual resolution of the acceleration signal is, however, limited by the time resolution of the counting devices used to decode the duty cycle. The faster the counter clock, the higher the resolution of the duty cycle and the shorter the T2 period can be for a given resolution. The following table shows some of the trade-offs. It is important to note that this is the resolution due to the microprocessors' counter. It is probable that the accelerometer's noise floor may set the lower limit on the resolution, as discussed in the previous section.
Table V. Trade-Offs Between Microcontroller Counter Rate, T2 Period, and Resolution of Duty Cycle Modulator

<table>
<thead>
<tr>
<th>T2 (ms)</th>
<th>( R_{\text{SET}} ) (kΩ)</th>
<th>ADXL202E Sample Rate</th>
<th>Counter-Clock Rate (MHz)</th>
<th>Counts per T2 Cycle</th>
<th>Counts per g</th>
<th>Resolution (mg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>124</td>
<td>1000</td>
<td>2.0</td>
<td>2000</td>
<td>250</td>
<td>4.0</td>
</tr>
<tr>
<td>1.0</td>
<td>124</td>
<td>1000</td>
<td>1.0</td>
<td>1000</td>
<td>125</td>
<td>8.0</td>
</tr>
<tr>
<td>1.0</td>
<td>124</td>
<td>1000</td>
<td>0.5</td>
<td>500</td>
<td>62.5</td>
<td>16.0</td>
</tr>
<tr>
<td>5.0</td>
<td>625</td>
<td>200</td>
<td>2.0</td>
<td>10000</td>
<td>1250</td>
<td>0.8</td>
</tr>
<tr>
<td>5.0</td>
<td>625</td>
<td>200</td>
<td>1.0</td>
<td>5000</td>
<td>625</td>
<td>1.6</td>
</tr>
<tr>
<td>5.0</td>
<td>625</td>
<td>200</td>
<td>0.5</td>
<td>2500</td>
<td>312.5</td>
<td>3.2</td>
</tr>
<tr>
<td>10.0</td>
<td>1250</td>
<td>100</td>
<td>2.0</td>
<td>20000</td>
<td>2500</td>
<td>0.4</td>
</tr>
<tr>
<td>10.0</td>
<td>1250</td>
<td>100</td>
<td>1.0</td>
<td>10000</td>
<td>1250</td>
<td>0.8</td>
</tr>
<tr>
<td>10.0</td>
<td>1250</td>
<td>100</td>
<td>0.5</td>
<td>5000</td>
<td>625</td>
<td>1.6</td>
</tr>
</tbody>
</table>

STRATEGIES FOR USING THE DUTY CYCLE OUTPUT WITH MICROCONTROLLERS
Application notes outlining various strategies for using the duty cycle output with low cost microcontrollers are available from the factory.

USING THE ADXL202E AS A DUAL-AXIS TILT SENSOR
One of the most popular applications of the ADXL202E is tilt measurement. An accelerometer uses the force of gravity as an input vector to determine orientation of an object in space.

An accelerometer is most sensitive to tilt when its sensitive axis is perpendicular to the force of gravity, i.e., parallel to the earth’s surface. At this orientation its sensitivity to changes in tilt is highest. When the accelerometer is oriented on axis to gravity, i.e., near its +1 g or -1 g reading, the change in output acceleration per degree of tilt is negligible. When the accelerometer is perpendicular to gravity, its output will change nearly 17.5 mg per degree of tilt, but at 45° degrees it is changing only at 12.2 mg per degree and resolution declines. The following table illustrates the changes in the X and Y axes as the device is tilted ±90° through gravity.

A DUAL AXIS TILT SENSOR: CONVERTING ACCELERATION TO TILT
When the accelerometer is oriented so both its X and Y axes are parallel to the earth’s surface it can be used as a two axis tilt sensor with a roll and a pitch axis. Once the output signal from the accelerometer has been converted to an acceleration that varies between -1 g and +1 g, the output tilt in degrees is calculated as follows:

\[
\text{Pitch} = \text{ASIN} (Ax/1 \, \text{g}) \\
\text{Roll} = \text{ASIN} (Ay/1 \, \text{g})
\]

Be sure to account for overranges. It is possible for the accelerometers to output a signal greater than ±1 g due to vibration, shock or other accelerations.

MEASURING 360° OF TILT
It is possible to measure a full 360° of orientation through gravity by using two accelerometers oriented perpendicular to one another (see Figure 5). When one sensor is reading a maximum change in output per degree, the other is at its minimum.

![Figure 5. Using a Two-Axis Accelerometer to Measure 360° of Tilt](image)

USING THE ANALOG OUTPUT
The ADXL202E was specifically designed for use with its digital outputs, but has provisions to provide analog outputs as well.

Duty Cycle Filtering
An analog output can be reconstructed by filtering the duty cycle output. This technique requires only passive components. The duty cycle period (T2) should be set to <1 ms. An RC filter with a 3 dB point at least a factor of >10 less than the duty cycle frequency is connected to the duty cycle output. The filter resistor should be no less than 100 kΩ to prevent loading of the output stage. The analog output signal will be ratiometric to the supply voltage. The advantage of this method is an output scale factor of approximately double the analog output. Its disadvantage is that the frequency response will be lower than when using the \( X_{\text{FILT}} \) and \( Y_{\text{FILT}} \) output.

\( X_{\text{FILT}}, Y_{\text{FILT}} \) Output
The second method is to use the analog output present at the \( X_{\text{FILT}} \) and \( Y_{\text{FILT}} \) pin. Unfortunately, these pins have a 32 kΩ output impedance and are not designed to drive a load directly. An op amp follower may be required to buffer this pin. The advantage of this method is that the full 5 kHz bandwidth of the accelerometer is available to the user. A capacitor still must be added at this point for filtering. The duty cycle converter should be kept running by using \( R_{\text{SET}} <10 \, \text{MΩ} \). Note that the accelerometer offset and sensitivity are ratiometric to the supply voltage. The offset and sensitivity are nominally:

\[
0 \, \text{g Offset} = V_{\text{DD}/2}
\]

ADXL202E Sensitivity = \((60 \, \text{mV} \times V_{\text{G}})/g\)
ADXL202E

USING THE ADXL202E IN VERY LOW POWER APPLICATIONS

An application note outlining low power strategies for the ADXL202E is available. Some key points are presented here. It is possible to reduce the ADXL202E's average current from 0.6 mA to less than 20 μA by using the following techniques:

1. Power Cycle the accelerometer.
2. Run the accelerometer at a Lower Voltage, (Down to 3 V).

Power Cycling with an External A/D

Depending on the value of the X_FILT capacitor, the ADXL202E is capable of turning on and giving a good reading in 1.6 ms. Most microcontroller based A/Ds can acquire a reading in another 25 μs. Thus it is possible to turn on the ADXL202E and take a reading in <2 ms. If we assume that a 20 Hz sample rate is sufficient, the total current required to take 20 samples is 2 ms × 20 samples/s × 0.6 mA = 24 μA average current. Running the part at 3 V will reduce the supply current from 0.6 mA to 0.4 mA, bringing the average current down to 16 μA.

The A/D should read the analog output of the ADXL202E at the X_FILT and Y_FILT pins. A buffer amplifier is recommended, and may be required in any case to amplify the analog output to give enough resolution with an 8-bit to 10-bit converter.

Power Cycling When Using the Digital Output

An alternative is to run the microcontroller at a higher clock rate and put it into shutdown between readings, allowing the use of the digital output. In this approach the ADXL202E should be set at its fastest sample rate (T2 = 0.5 ms), with a 500 Hz filter at X_FILT and Y_FILT. The concept is to acquire a reading as quickly as possible and then shut down the ADXL202E and the microcontroller until the next sample is needed.

In either of the above approaches, the ADXL202E can be turned on and off directly using a digital port pin on the microcontroller to power the accelerometer without additional components.

CALIBRATING THE ADXL202E/ADXL210

The initial value of the offset and scale factor for the ADXL202E will require calibration for applications such as tilt measurement. The ADXL202E architecture has been designed so that these calibrations take place in the software of the microcontroller used to decode the duty cycle signal. Calibration factors can be stored in EEPROM or determined at turn-on and saved in dynamic memory.

For low g applications, the force of gravity is the most stable, accurate and convenient acceleration reference available. A reading of the 0 g point can be determined by orientating the device parallel to the earth’s surface and then reading the output.

A more accurate calibration method is to make measurements at +1 g and -1 g. The sensitivity can be determined by the two measurements.

To calibrate, the accelerometer's measurement axis is pointed directly at the earth. The 1 g reading is saved and the sensor is turned 180° to measure -1 g. Using the two readings, the sensitivity is:

Let \( A = \) Accelerometer output with axis oriented to +1 g
Let \( B = \) Accelerometer output with axis oriented to -1 g then:
Sensitivity = \( \frac{A - B}{2} \)

For example, if the +1 g reading (A) is 55% duty cycle and the -1 g reading (B) is 32% duty cycle, then:
Sensitivity = \( \frac{55\% - 32\%}{2} \) g = 11.5%/g

These equations apply whether the output is analog or duty cycle. Application notes outlining algorithms for calculating acceleration from duty cycle and automated calibration routines are available from the factory.

OUTLINE DIMENSIONS

Dimensions shown in inches and (mm).

8-Terminal Ceramic Leadless Chip Carrier (E-8)

CONTROLLING DIMENSIONS ARE IN MILLIMETERS
±150°/s Single Chip Yaw Rate Gyro with Signal Conditioning

FEATURES
Complete rate gyroscope on a single chip
Z-axis (yaw rate) response
High vibration rejection over wide frequency
0.05°/s/√Hz noise
2000 g powered shock operation
Self-test on digital command
Temperature sensor output
Precision voltage reference output
Absolute rate output for precision applications
5 V single-supply operation
Ultrasmall and light (< 0.15 cc, < 0.5 gram)

APPLICATIONS
GPS navigation systems
Vehicle stability control
Inertial measurement units
Guidance and control
Platform stabilization

GENERAL DESCRIPTION
The ADXRS150 is a complete angular rate sensor (gyroscope) that uses Analog Devices' surface-micromachining process to make a functionally complete and low cost angular rate sensor integrated with all of the required electronics on one chip.

The manufacturing technique for this device is the same high volume BIMOS process used for high reliability automotive airbag accelerometers.

The output signal, RATEOUT (1B, 2A), is a voltage proportional to the angular rate about the axis normal to the top surface of the package (see Figure 2). A single external resistor can be used to lower the scale factor. An external capacitor is used to set the bandwidth. Other external capacitors are required for operation (see Figure 21).

A precision reference and a temperature output are also provided for compensation techniques. Two digital self-test inputs electromechanically excite the sensor to test the operation of both sensors and the signal conditioning circuits. The ADXRS150 is available in a 7 mm x 7 mm x 3 mm BGA surface-mount package.

FUNCTIONAL BLOCK DIAGRAM

Rev. A
Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective companies.
# ADXRS150

## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADXRS150—Specifications</td>
<td>3</td>
</tr>
<tr>
<td>Absolute Maximum Ratings</td>
<td>4</td>
</tr>
<tr>
<td>Rate Sensitive Axis</td>
<td>4</td>
</tr>
<tr>
<td>Typical Performance Characteristics</td>
<td>5</td>
</tr>
<tr>
<td>Theory of Operation</td>
<td>8</td>
</tr>
<tr>
<td>Supply and Common Considerations</td>
<td>8</td>
</tr>
<tr>
<td>Setting Bandwidth</td>
<td>8</td>
</tr>
<tr>
<td>Increasing Measurement Range</td>
<td>9</td>
</tr>
<tr>
<td>Temperature Output and Calibration</td>
<td>9</td>
</tr>
<tr>
<td>Using the ADXRS150 with a Supply-Ratiometric ADC</td>
<td>9</td>
</tr>
<tr>
<td>Null Adjustment</td>
<td>9</td>
</tr>
<tr>
<td>Self-Test Function</td>
<td>9</td>
</tr>
<tr>
<td>Continuous Self-Test</td>
<td>9</td>
</tr>
<tr>
<td>Acceleration Sensitivity</td>
<td>10</td>
</tr>
<tr>
<td>Pin Configurations And Functional Descriptions</td>
<td>11</td>
</tr>
<tr>
<td>Outline Dimensions</td>
<td>12</td>
</tr>
</tbody>
</table>

## REVISION HISTORY

Revision A

1/03—Data Sheet Changed from REV. 0 to REV. A

Edit to Figure 5

---

5
# ADXRS150 — SPECIFICATIONS

Table 1. @\(T_A = 25^\circ C\), \(V_S = 5\) V, Bandwidth = 80 Hz (\(C_{OUT} = 0.01\) \(\mu F\)), Angular Rate = 0°/s, Unless Otherwise Noted.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Conditions</th>
<th>ADXRS150ABG</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sensitivity</strong></td>
<td>Dynamic Range&lt;sup&gt;2&lt;/sup&gt;, Clockwise Rotation Is Positive Output, Full-Scale Range over Specifications Range</td>
<td>Min&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
<tr>
<td>Initial</td>
<td>@25°C</td>
<td>(\pm 150)</td>
</tr>
<tr>
<td>Over Temperature</td>
<td>(V_{CC} = 4.75) V to 5.25 V</td>
<td>11.25</td>
</tr>
<tr>
<td>Nonlinearity</td>
<td>Best Fit Straight Line</td>
<td>0.7</td>
</tr>
<tr>
<td>Voltage Sensitivity</td>
<td>(V_{CC} = 4.75) V to 5.25 V</td>
<td>0.7</td>
</tr>
<tr>
<td><strong>NULL</strong></td>
<td>Initial Null</td>
<td>2.50</td>
</tr>
<tr>
<td>Null Drift over Temperature&lt;sup&gt;3&lt;/sup&gt;</td>
<td>Delta from 25°C</td>
<td>(\pm 300)</td>
</tr>
<tr>
<td>Turn-On Time</td>
<td>Power on to (\pm 1/2)°/s of Final</td>
<td>0.2</td>
</tr>
<tr>
<td>Linear Acceleration Effect</td>
<td>Any Axis</td>
<td>1</td>
</tr>
<tr>
<td>Voltage Sensitivity</td>
<td>(V_{CC} = 4.75) V to 5.25 V</td>
<td>50</td>
</tr>
<tr>
<td><strong>Noise Performance</strong></td>
<td>Rate Noise Density</td>
<td>@25°C</td>
</tr>
<tr>
<td><strong>Frequency Response</strong></td>
<td>3 dB Bandwidth&lt;sup&gt;4&lt;/sup&gt; (User Selectable)</td>
<td>22 nF as Comp Cap (See Applications section)</td>
</tr>
<tr>
<td>Sensor Resonant Frequency</td>
<td></td>
<td>14</td>
</tr>
<tr>
<td><strong>Self Test</strong></td>
<td>ST1 RATEOUT Response&lt;sup&gt;5&lt;/sup&gt;</td>
<td>ST1 Pin from Logic “0” to “1,” –40°C to +85°C</td>
</tr>
<tr>
<td>ST2 RATEOUT Response&lt;sup&gt;5&lt;/sup&gt;</td>
<td>ST2 Pin from Logic “0” to “1,” –40°C to +85°C</td>
<td>+400</td>
</tr>
<tr>
<td>Logic “1” Input Voltage</td>
<td>Standard High Logic Level Definition</td>
<td>3.3</td>
</tr>
<tr>
<td>Logic “0” Input Voltage</td>
<td>Standard Low Logic Level Definition</td>
<td>1.7</td>
</tr>
<tr>
<td>Input Impedance</td>
<td>To Common</td>
<td>50</td>
</tr>
<tr>
<td><strong>Temperature Sensor</strong></td>
<td>(V_{OUT}) at 298°C</td>
<td>2.50</td>
</tr>
<tr>
<td>Max Current Load on Pin</td>
<td>Source to Common</td>
<td>50</td>
</tr>
<tr>
<td>Scale Factor</td>
<td>Proportional to Absolute Temperature</td>
<td>8.4</td>
</tr>
<tr>
<td><strong>Output Drive Capability</strong></td>
<td>Output Voltage Swing</td>
<td>(I_{OUT} = \pm 100) μA</td>
</tr>
<tr>
<td>Capacitive Load Drive</td>
<td></td>
<td>1000</td>
</tr>
<tr>
<td><strong>2.5 V Reference</strong></td>
<td>Voltage Value</td>
<td>2.45</td>
</tr>
<tr>
<td>Load Drive to Ground</td>
<td>Source</td>
<td>2.5</td>
</tr>
<tr>
<td>Load Regulation</td>
<td>0 &lt; (I_{OUT}) &lt; 200 μA</td>
<td>200</td>
</tr>
<tr>
<td>Power Supply Rejection</td>
<td>(V_S) to 5.25 (V_S)</td>
<td>5.0</td>
</tr>
<tr>
<td>Temperature Drift&lt;sup&gt;4&lt;/sup&gt;</td>
<td>Delta from 25°C</td>
<td>1.0</td>
</tr>
<tr>
<td><strong>Power Supply</strong></td>
<td>Operating Voltage Range</td>
<td></td>
</tr>
<tr>
<td>Quiescent Supply Current</td>
<td></td>
<td>5.00</td>
</tr>
<tr>
<td><strong>Temperature Range</strong></td>
<td>Specified Performance Grade A</td>
<td>–40</td>
</tr>
</tbody>
</table>

<sup>1</sup> All min and max specifications are guaranteed. Typical specifications are not tested or guaranteed.
<sup>2</sup> Dynamic range is the maximum full-scale measurement range possible, including output swing range, initial offset, sensitivity, offset drift, and sensitivity drift at 5 V supplies.
<sup>3</sup> Specification refers to the maximum extent of this parameter as a worst-case value at \(T_{Min}\) or \(T_{Max}\).
<sup>4</sup> Frequency at which response is 3 dB down from dc response with specified compensation capacitor value. Internal pole forming resistor is 180 kΩ. See Setting Bandwidth section.
<sup>5</sup> Self-test response varies with temperature. Refer to the Self-Test Function section for details.
**Absolute Maximum Ratings**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceleration (Any Axis, Unpowered, 0.5 ms)</td>
<td>2000 g</td>
</tr>
<tr>
<td>Acceleration (Any Axis, Powered, 0.5 ms)</td>
<td>2000 g</td>
</tr>
<tr>
<td>$+V_S$</td>
<td>$-0.3 \text{ V to } +6.0 \text{ V}$</td>
</tr>
<tr>
<td>Output Short-Circuit Duration (Any Pin to Common)</td>
<td>Indefinite</td>
</tr>
<tr>
<td>Operating Temperature Range</td>
<td>$-55^\circ \text{C to } +125^\circ \text{C}$</td>
</tr>
<tr>
<td>Storage Temperature</td>
<td>$-65^\circ \text{C to } +150^\circ \text{C}$</td>
</tr>
</tbody>
</table>

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Applications requiring more than 200 cycles to MIL-STD-883 Method 1010 Condition B ($-55^\circ \text{C to } +125^\circ \text{C}$) require underfill or other means to achieve this requirement.

Drops onto hard surfaces can cause shocks of greater than 2000 g and exceed the absolute maximum rating of the device. Care should be exercised in handling to avoid damage.

**Rate Sensitive Axis**

This is a Z-axis rate-sensing device that is also called a yaw-rate sensing device. It produces a positive going output voltage for clockwise rotation about the axis normal to the package top, i.e., clockwise when looking down at the package lid.
TYPICAL PERFORMANCE CHARACTERISTICS

Figure 3. Rate Sensing Start-Up Time

Figure 4. Null Stability for 1 Hour

Figure 5. Temperature Sensor Output

Figure 6. Null Settling Time

Figure 7. Root Allan Variance vs. Averaging Time

Figure 8. 2.5 V Voltage Reference vs. Temperature
ADXRS150 @ BW = 40 Hz, Typical Vibration Characteristics, 10 g Flat Band, 20 Hz to 2 kHz

Figure 9. 10 g Random Vibration in Package-Lateral Axis Orientation

Figure 10. 10 g Random Vibration in Package-Longitudinal Axis Orientation

Figure 11. 10 g Random Vibration in Rate Axis Orientation

Figure 12. 10 g Random Vibration in Package-Lateral Axis Orientation (0.5s Average)

Figure 13. 10 g Random Vibration in Package-Longitudinal Axis Orientation (0.5s Average)

Figure 14. 10 g Random Vibration in Rate Axis Orientation
Behavior under Various Shock Test Conditions

Figure 15. Shock Test 100 g, 5 ms in Lateral Axis (40 Hz)

Figure 18. Shock Test 100 g, 5 ms in Longitudinal Axis (40 Hz)

Figure 16. Hi-g Shock Test in Lateral Axis (40 Hz)

Figure 19. Hi-g Shock Test, Lateral Axis, 10x Time Base (40 Hz)

Figure 17. Hi-g Shock in Rate Axis (40 Hz)

Figure 20. Hi-g Shock, Rate Axis, BW Reduced to 8 Hz
THEORY OF OPERATION

The ADXRS150 operates on the principle of a resonator gyro. Two polysilicon sensing structures each contain a dither frame, which is electrostatically driven to resonance. This produces the necessary velocity element to produce a Coriolis force during angular rate. At two of the outer extremes of each frame, orthogonal to the dither motion, are movable fingers that are placed between fixed pickoff fingers to form a capacitive pickoff structure that senses Coriolis motion. The resulting signal is fed to a series of gain and demodulation stages that produce the electrical rate signal output. The dual-sensor design rejects external g-forces and vibration. Fabricating the sensor with the signal conditioning electronics preserves signal integrity in noisy environments.

The electrostatic resonator requires 14 V to 16 V for operation. Since only 5 V is typically available in most applications, a charge pump is included on-chip. If an external 14 V to 16 V supply is available, the two capacitors on CP1–CP4 can be omitted and this supply can be connected to CP5 (Pin 7D) with a 100 nF decoupling capacitor in place of the 47 nF.

After the demodulation stage there is a single-pole low-pass filter consisting of an internal 9 kΩ resistor (RSENS) and an external user-supplied capacitor (CMID). A CMID capacitor of 100 nF sets a 400 Hz ±35% low-pass pole and is used to limit high frequency artifacts before final amplification. Bandwidth limit capacitor, COUT, sets the pass bandwidth (see Figure 22 and the Setting Bandwidth section).

![Figure 21. Example Application Circuit (Top View)](image)

NOTE THAT INNER ROWS/COLUMNS OF PINS HAVE BEEN OMITTED FOR CLARITY BUT SHOULD BE CONNECTED IN THE APPLICATION.

Supply and Common Considerations

Only power supplies used for supplying analog circuits are recommended for powering the ADXRS150. High frequency noise and transients associated with digital circuit supplies may have adverse effects on device operation.

Figure 21 shows the recommended connections for the ADXRS150 where both AVCC and PDD have a separate decoupling capacitor. These should be placed as close to their respective pins as possible before routing to the system analog supply. This will minimize the noise injected by the charge pump that uses the PDD supply.

It is also recommended to place the charge pump capacitors connected to the CP1–CP4 pins as close to the part as possible. These capacitors are used to produce the on-chip high voltage supply switched at the dither frequency at approximately 14 kHz. Care should be taken to ensure that there is no more than 50 pF of stray capacitance between CP1–CP4 and ground. Surface-mount chip capacitors are suitable as long as they are rated for over 15 V.

![Figure 22. Block Diagram with External Components](image)

**Setting Bandwidth**

External capacitors CMID and COUT are used in combination with on-chip resistors to create two low-pass filters to limit the bandwidth of the ADXRS150’s rate response. The –3 dB frequency set by Rout and COUT is:

\[
\omega_{3dB} = \frac{1}{\sqrt{2\pi \times Rout \times COUT}}
\]

and can be well controlled since Rout has been trimmed during manufacturing to be 180 kΩ ±1%. Any external resistor applied between the RATEOUT (1B, 2A) and SUMJ (1C, 2C) pins will result in:

\[
R_{OUT} = \left(\frac{180 \text{kΩ} \times R_{EXT}}{180 \text{kΩ} + R_{EXT}}\right)
\]

The –3 dB frequency is set by RSENS (the parallel combination...
of $R_{SEN}$ and $R_{SEN}$) at about 4.5 kΩ nominal, and CMID is less well controlled since $R_{SEN}$ and $R_{SEN}$ have been used to trim the rate sensitivity during manufacturing and have a ±35% tolerance. Its primary purpose is to limit the high frequency demodulation artifacts from saturating the final amplifier stage. Thus, this pole of nominally 400 Hz @ 0.1 μF need not be precise. Lower frequency is preferable, but its variability usually requires it to be about 10 times greater (in order to preserve phase integrity) than the well-controlled output pole. In general, both -3 dB filter frequencies should be set as low as possible to reduce the amplitude of these high frequency artifacts as well as to reduce the overall system noise.

Increasing Measurement Range

The full-scale measurement range of the ADXRS150 can be increased by placing an external resistor between the RATEOUT (1B, 2A) and SUMJ (1C, 2C) pins, which would parallel the internal $R_{OUT}$ resistor that is factory-trimmed to 180 kΩ. For example, a 330 kΩ external resistor will give approximately 8.1 mV/°/sec sensitivity and a commensurate ~50% increase in the full-scale range. This is effective for up to a 4× increase in the full-scale range (minimum value of the parallel resistor allowed is 45 kΩ). Beyond this amount of external sensitivity reduction, the internal circuitry headroom requirements prevent further increase in the linear full-scale output range. The drawbacks of modifying the full-scale range are the additional output null drift (as much as 2%/sec over temperature) and the readjustment of the initial null bias (see the Null Adjust section).

Temperature Output and Calibration

It is common practice to temperature-calibrate gyro's to improve their overall accuracy. The ADXRS150 has a temperature-proportional voltage output that provides input to such a calibration method. The voltage at TEMP (3F, 3G) is nominally 2.5 V at 27°C and has a PTAT (proportional to absolute temperature) characteristic of 8.4 mV/°C. Note that the TEMP output circuitry is limited to 50 μA source current.

Using a 3-point calibration technique, it is possible to calibrate the ADXRS150's null and sensitivity drift to an overall accuracy of nearly 300/°/hour. An overall accuracy of 70/°/hour or better is possible using more points. Limiting the bandwidth of the device reduces the flat-band noise during the calibration process, improving the measurement accuracy at each calibration point.

Using the ADXRS150 with a Supply-Ratiometric ADC

The ADXRS150's RATEOUT signal is nonratiometric, i.e., neither the null voltage nor the rate sensitivity is proportional to the supply. Instead they are nominally constant for dc supply changes within the 4.75 V to 5.25 V operating range. If the ADXRS150 is used with a supply-ratiometric ADC, the ADXRS150's 2.5 V output can be converted and used to make corrections in software for the supply variations.

Null Adjustment

Null adjustment is possible by injecting a suitable current to SUMJ (1C, 2C). Adding a suitable resistor to either ground or the positive supply is a simple way of achieving this. The nominal 2.5 V null is for a symmetrical swing range at RATEOUT (1B, 2A). However, a nonsymmetrical output swing may be suitable in some applications. Note that if a resistor is connected to the positive supply, supply disturbances may reflect some null instability. Digital supply noise should be avoided particularly in this case (see Supply and Common Considerations section). The resistor value to use is approximately:

$$R_{NULL} = \frac{2.5 \times 180,000}{(V_{NULL} - V_{NULLI})}$$

$V_{NULL}$ is the unadjusted zero rate output, and $V_{NULLI}$ is the target null value. If the initial value is below the desired value, the resistor should terminate on common or ground. If it is above the desired value, the resistor should terminate on the 5 V supply. Values typically are in the 1 MΩ to 5 MΩ range.

If an external resistor is used across RATEOUT and SUMJ, then the parallel equivalent value is substituted into the above equation. Note that the resistor value is an estimate since it assumes $V_{CC} = 5.0 \text{ V}$ and $V_{SUM} = 2.5 \text{ V}$.

Self-Test Function

The ADXRS150 includes a self-test feature that activates each of the sensing structures and associated electronics in the same manner as if subjected to angular rate. It is activated by standard logic high levels applied to inputs ST1 (5F, 5G), ST2 (4F, 4G), or both. ST1 will cause the voltage at RATEOUT to change about -0.66 V and ST2 will cause an opposite change of +0.66 V. The self-test response follows the viscosity temperature dependence of the package atmosphere, approximately 0.25%/°C.

Activating both ST1 and ST2 simultaneously is not damaging. Since ST1 and ST2 are not necessarily closely matched, actuating both simultaneously may result in an apparent null bias shift.

Continuous Self-Test

The one-chip integration of the ADXRS150 gives it higher reliability than is obtainable with any other high volume manufacturing method. Also, it is manufactured under a mature BIMOS process that has field-proven reliability. As an additional failure detection measure, power-on self-test can be performed. However, some applications may warrant continuous self-test while
sensing rate. Application notes outlining continuous self-test techniques are also available on the Analog Devices website.

**Acceleration Sensitivity**

The sign convention used is that lateral acceleration is positive in the direction from Pin Column A to Pin Column G of the package. That is, a device has positive sensitivity if its voltage output increases when the row of Pins 2A–6A are tipped under the row of Pins 2G–6G in the earth’s gravity.

There are two effects of concern, shifts in the static null and induced null noise. Scale factor is not significantly affected until the acceleration reaches several hundred m/s².

Vibration rectification for frequencies up to 20 kHz is on the order of 0.00002(°/s)/(m/s²), is not significantly dependent on frequency, and has been verified up to 400 m/s² rms.

Linear vibration spectral density near the 14 kHz sensor resonance translates into output noise. In order to have a significant effect, the vibration must be within the angular rate bandwidth (typically ±40 Hz of the resonance), so it takes considerable high frequency vibration to have any effect.

Away from the 14 kHz resonance the effect is not discernible, except for vibration frequencies within the angular rate pass band. This can be seen in Figure 9 to Figure 14 for the various sensor axes. The in-band effect can be seen in Figure 24. This is the result of the static g-sensitivity. The specimen used for Figure 24 had a g-sensitivity of 0.15°/s/g and its total in-band noise degraded from 3 mV rms to 5 mV rms for the specified vibration. The effect of broadband vibration up to 20 kHz is shown in Figure 23 and Figure 25.

The output noise of the part falls away in accordance with the output low-pass filter and does not contain any "spikes" greater than 1% of the low frequency noise. A typical noise spectrum is shown in Figure 26.

![Figure 23. Random Vibration (Lateral) 10 kHz to 20 kHz at 0.01 g/\(\sqrt{Hz}\) with 60 Hz Sampling and 0.5 Sec Averaging](image)

![Figure 24. Random Vibration (Lateral) 2 Hz to 40 Hz, 3.2 g rms](image)

![Figure 25. Random Vibration (Lateral) 10 kHz to 20 kHz at 0.01 g/\(\sqrt{Hz}\) with 60 Hz Sampling and 0.5 Sec Averaging](image)

![Figure 26. Noise Spectral Density at RATEOUT–BW = 4Hz](image)
**PIN CONFIGURATIONS AND FUNCTIONAL DESCRIPTIONS**

![Diagram of pin configurations](image)

Figure 27. BGA-32 (Bottom View)

Table 3. Pin Function Descriptions—32-Lead BGA

<table>
<thead>
<tr>
<th>Pin No.</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6D, 7D</td>
<td>CP5</td>
<td>HV Filter Capacitor—47 nF</td>
</tr>
<tr>
<td>6A, 7B</td>
<td>CP4</td>
<td>Charge Pump Capacitor—22 nF</td>
</tr>
<tr>
<td>6C, 7C</td>
<td>CP3</td>
<td>Charge Pump Capacitor—22 nF</td>
</tr>
<tr>
<td>5A, 5B</td>
<td>CP1</td>
<td>Charge Pump Capacitor—22 nF</td>
</tr>
<tr>
<td>4A, 4B</td>
<td>CP2</td>
<td></td>
</tr>
<tr>
<td>3A, 3B</td>
<td>AVCC</td>
<td>+ Analog Supply</td>
</tr>
<tr>
<td>1B, 2A</td>
<td>RATEOUT</td>
<td>Rate Signal Output</td>
</tr>
<tr>
<td>1C, 2C</td>
<td>SUMJ</td>
<td>Output Amp Summing Junction</td>
</tr>
<tr>
<td>1D, 2D</td>
<td>CMID</td>
<td>HF Filter Capacitor—100 nF</td>
</tr>
<tr>
<td>1E, 2E</td>
<td>2.5V</td>
<td>2.5 V Precision Reference</td>
</tr>
<tr>
<td>1F, 2G</td>
<td>AGND</td>
<td>Analog Supply Return</td>
</tr>
<tr>
<td>3F, 3G</td>
<td>TEMP</td>
<td>Temperature Voltage Output</td>
</tr>
<tr>
<td>4F, 4G</td>
<td>ST2</td>
<td>Self-Test for Sensor 2</td>
</tr>
<tr>
<td>5F, 5G</td>
<td>ST1</td>
<td>Self-Test for Sensor 1</td>
</tr>
<tr>
<td>6G, 7F</td>
<td>PGND</td>
<td>Charge Pump Supply Return</td>
</tr>
<tr>
<td>6E, 7E</td>
<td>PDD</td>
<td>+ Charge Pump Supply</td>
</tr>
</tbody>
</table>
**ESD CAUTION**

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although this product features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

**Table 4. Ordering Guide**

<table>
<thead>
<tr>
<th>ADXRS150 Products</th>
<th>Temperature Package</th>
<th>Package Description</th>
<th>Package Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADXRS150ABG</td>
<td>-40°C to +85°C</td>
<td>32-Lead BGA</td>
<td>BC-32</td>
</tr>
<tr>
<td>ADXRS150ABG-Reel</td>
<td>-40°C to +85°C</td>
<td>32-Lead BGA</td>
<td>BC-32</td>
</tr>
</tbody>
</table>