Using Embedded Systems to Determine the Configuration of a Static Wheelchair Mounted Robotic Arm

by

Daniel Ryan Ashley

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Engineering
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Co-Major Professor: Redwan Alqasemi, Ph.D.
Co-Major Professor: Srinivas Katkoori, Ph.D.
Luther Palmer, Ph.D.

Date of Approval: October 30, 2014

Keywords: WMRA, kinematics, calibration, robotic manipulator

Copyright © 2014, Daniel Ryan Ashley
ACKNOWLEDGMENTS

I would like to thank Dr. Alqasemi for being an extremely helpful adviser. Without the opportunities provided by him, I would not have pursued a graduate degree. His insight and help with my work has been invaluable.

I would also like to thank Dr. Katkoori and Dr. Palmer for their helpful insights during this project.

I would like to thank Dr. Rajiv Dubey, the opportunities provided by him have helped further my career.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>iii</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>iv</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>v</td>
</tr>
<tr>
<td>CHAPTER 1</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Raspberry Pi</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Inertial Measurement Unit</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Accelerometers</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Thesis Organization</td>
<td>3</td>
</tr>
<tr>
<td>CHAPTER 2</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Related Work</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Key Points</td>
<td>7</td>
</tr>
<tr>
<td>2.3 Summary</td>
<td>8</td>
</tr>
<tr>
<td>CHAPTER 3</td>
<td>9</td>
</tr>
<tr>
<td>3.1 Raspberry Pi</td>
<td>9</td>
</tr>
<tr>
<td>3.2 Inertial Measurement Unit</td>
<td>10</td>
</tr>
<tr>
<td>3.3 Accelerometers</td>
<td>10</td>
</tr>
<tr>
<td>CHAPTER 4</td>
<td>13</td>
</tr>
<tr>
<td>4.1 Orientation</td>
<td>14</td>
</tr>
<tr>
<td>4.2 Cardinal Direction</td>
<td>15</td>
</tr>
<tr>
<td>CHAPTER 5</td>
<td>16</td>
</tr>
<tr>
<td>5.1 Joints 1 and 2</td>
<td>19</td>
</tr>
<tr>
<td>5.2 Joints 3 and 4</td>
<td>21</td>
</tr>
<tr>
<td>5.3 Joints 5 and 6</td>
<td>23</td>
</tr>
<tr>
<td>5.4 Joint 7</td>
<td>25</td>
</tr>
<tr>
<td>CHAPTER 6</td>
<td>28</td>
</tr>
<tr>
<td>6.1 Pitch and Roll Error</td>
<td>28</td>
</tr>
<tr>
<td>6.2 Cumulative Error</td>
<td>30</td>
</tr>
<tr>
<td>6.3 Configuration Error</td>
<td>30</td>
</tr>
<tr>
<td>CHAPTER 7</td>
<td>31</td>
</tr>
<tr>
<td>7.1 Testing Procedure</td>
<td>31</td>
</tr>
<tr>
<td>7.2 Data and Results</td>
<td>31</td>
</tr>
<tr>
<td>APPENDICES</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>Appendix A    IEEE</td>
<td>43</td>
</tr>
<tr>
<td>Appendix B    Characterizing Suitability of Wearable Sensors for Movement Analysis Using a Programmed Robotic Motion</td>
<td>44</td>
</tr>
<tr>
<td>Appendix C    Maximizing Manipulation Capabilities of Persons with Disabilities Using a Smart 9-Degree-of-Freedom Wheelchair-mounted Robotic Arm System</td>
<td>45</td>
</tr>
<tr>
<td>Appendix D    Freescale Application Note</td>
<td>46</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 5.1  Manipulator arm DH-parameters [1]  18
LIST OF FIGURES

Figure 2.1 Arm calibration using single three axis accelerometer [2] 5
Figure 2.2 WMRA with reflectors attached for Vicon system [3] 7
Figure 3.1 ADXL 345 accelerometer using SPI 4-wire communication 11
Figure 3.2 ADXL 345 accelerometer using I2C communication 12
Figure 3.3 Wiring diagram for the embedded devices 12
Figure 4.1 WMRA base frame [1] 13
Figure 5.1 Manipulator arm [1] 19
Figure 5.2 Accelerometer one frame to joint two frame 20
Figure 5.3 Accelerometer two frame to joint four frame 22
Figure 5.4 Accelerometer three frame to joint six frame 24
Figure 5.5 Accelerometer four frame to joint seven frame 26
Figure 6.1 Pitch and roll error range [4]. 29
Figure 7.1 Joints 1 and 2 estimations using a 3 x 3 identity matrix as $R_W^0$ 32
Figure 7.2 Joints 1 and 2 estimations using computed $R_W^0$ from IMU data 33
Figure 7.3 Joints 1 through 4, with joint 1 moving through gravity vector alignment 34
Figure 7.4 Joints 5 through 7, with joint 1 moving through gravity vector alignment 34
Figure 7.5 Joints 1 and 2 computed values 35
Figure 7.6 Joints 3 and 4 computed values 35
Figure 7.7 Joints 5 and 6 computed values 35
Figure 7.8 Joint 7 computed values 35
Figure 8.1 Flowchart showing the order in which joints 1 through 7 are solved 38
ABSTRACT

The calibration of a 9 degree of freedom (DOF) robotic manipulator using multiple three axis accelerometers and an embedded system will be accomplished in this work. The 9-DOF robotic system used in this study is a 7-DOF robotic arm attached to a 2-DOF power wheelchair. Combined they create a Wheelchair Mounted Robotic Arm (WMRA). The problem that will be solved by this thesis is the calibration of the robotic system during start up. The 7 DOF robotic arm is comprised of rotational joints only. These joints have dual channel encoders to determine the joint position, among other useful data. The problem with dual channel encoders is that when power to the encoders is turned off and the motor is moved, then the robot controller does not have accurate position data when the system is powered again. The proposed calibration method will find the angles of two joints per three axis accelerometer. Four separate accelerometers are mounted on different locations of the 7 DOF robotic arm to determine the arms joint values. To determine the orientation of the base frame, an inertial measurements unit (IMU) is mounted to the origin of the base frame. By using this system of accelerometers and inertial measurement unit, the WMRA can be completely calibrated during system start up. The results collected for this calibration method show joint estimations with an error of ±0.1 radians for each joint. The results also show an accumulation of error for joints that are farther from the base frame.
CHAPTER 1
INTRODUCTION

The calibration of robotic arms is not a trivial procedure. One key point to consider is that when the manipulator arm is first initialized, none of the current arm’s joint positions are known. The problem with this is any erroneous movement while the arm is not correctly calibrated could lead to injury of the user and damage to the robotic arm itself. Therefore, a method of calibration is needed to determine the current location of each joint without the need to move the arm itself.

The easiest calibration method is for each joint in the arm to be equipped with absolute encoders. Absolute encoders are a type of encoder attached to a rotatory motor shaft. Using this device after a gear system, the current angle of the motor shaft can be determined without saving any previous encoder states [5]. During the initialization, the current values of the absolute encoders will determine the angle of each joint. A critical problem arises with absolute encoders when a gear system is attached after the encoder device. By adding a gear system, the absolute encoders no longer have a 1:1 relation to the position of the joint (per revolution) [5]. This means there will be multiple joint locations the arm could be in based on the readings from the absolute encoders.

The robotic system used in this work uses dual channel encoders which are capable of determining the direction of rotation as well as the number of encoder counts traveled while in a powered state. Essentially, this gives a relative joint angle based on the starting point of the joint. This experiment is going to assume a gearing system is used on each joint and a calibration method is needed to initialize the joint angles to their respective current positions.

The proposed embedded system incorporated for calibration consists of the following: a Raspberry Pi, an Inertial Measurement Unit (IMU), and multiple three axis Accelerometers. Together, these devices are used to determine the orientation of each link of the robotic manipulator as well as the cardinal direction that the WMRA is facing. In later chapters it will become apparent that the cardinal direction information is not necessary when estimating the joint values of the manipulator
arm. It can be useful in localizing the WMRA when moving around an environment, but isn’t necessary when calibrating the robotic manipulator.

1.1 Raspberry Pi

The Raspberry Pi is an embedded system featuring an ARM processor [6]. In this project, it is used as the central computer to communicate with the accelerometers and the IMU devices. It is also used to process the data to determine the angle of all the WMRA joints. The Raspberry Pi is capable of both I2C and SPI communication protocols [6], both of which were used in this project. Chapter 3.1 discusses the reasons why both modes of serial communication are needed. The Raspberry Pi provides 3.3V supply voltage for the IMU and the multiple accelerometers. All of the code implemented on the Raspberry Pi is written in C++. The on board SPI library, bcm3546, is used to communicate with two of the four accelerometers mounted to the lower portion of the manipulator arm. The I2C library, wiring pi, is used to communicate with the remaining two accelerometers and the IMU device. These two libraries are necessary to access the general purpose input/output pins on the raspberry Pi. Under certain conditions, it may be necessary to communicate with the manipulator arm’s motor controller. The Raspberry Pi uses an Ethernet connection to communicate with the arm’s motor controller.

1.2 Inertial Measurement Unit

The Inertial Measurement Unit (IMU) is a device which uses different force sensing devices to determine the movement and orientation of the IMU device [7]. The IMU used is the MPU-9150. IMUs are used in many robotic devices ranging from space shuttles to cell phones. In this case, the IMU device will be used as the sensing device for the WMRA base frame. The wheelchair’s orientation will be detected using the IMU’s magnetometer as well as its three axis accelerometer. The IMU’s accelerometer will also detect the inclination of the wheelchair so the arm can be properly calibrated to the corrected wheelchair reference frame. Without this reference, this method of calibration could only be done while the WMRA is on a perfectly flat and horizontal surface. The IMU device is installed at the origin of the WMRA base frame.
1.3 Accelerometers

The accelerometers used in this study are Analog Devices ADXL345 3-Axis Digital Accelerometer [8]. This accelerometer was chosen because of its compact size, low power consumption, and high resolution measurements. The high resolution is the key feature that is the most beneficial to this work. The ADXL 345 offers 10-bit resolution at ±2g, this will allow for accurate tilt sensing at 4mg/LSB [8]. There will be four accelerometers on the arm, each separated by two rotational joints. The accelerometers will be mounted so that they do not interfere with the robotic arm’s workspace. The accelerometers are mounted inside the arm to limit the possibilities of the sensor being damaged as well as limit calibration problems.

1.4 Thesis Organization

In Chapter 2, similar calibration methods will be discussed. Each work has a drawback that makes the method unusable in calibrating the WMRA in this project. The hardware used in this work is discussed in Chapter 3. Chapter 4 discusses the calibration of the base frame of the manipulator arm using an inertial measurement unit. The method of calibrating the 7-DOF robotic manipulator is discussed in Chapter 5. The types of errors that occur when using this calibration method are discussed in Chapter 6. Chapter 7 shows the experimental results found when using the calibration method proposed in this work.
CHAPTER 2

BACKGROUND AND RELATED WORK

Calibration of a robotic system is a common problem to come across when working with almost any type of robotic platform. That being said, there are many methods to calibrate these systems. Some are robot specific, while others can be used or easily modified to work with other robotic platforms. For example, the Inertial Measurement Unit (IMU) used to determine the initial orientation of the WMRA could also be used on a mobile robot platform to calibrate the device’s cardinal direction [7]. In this chapter, we will only review the calibration methods that are closely related the WMRA system.

2.1 Related Work

One closely related method of manipulator arm calibration is using a single accelerometer mounted to the end-effector of the robotic arm. Each joint would perform a series of individual movements, while the other joints remain static. Using the accelerometer and the motor’s rotary encoders, the current joint angle can be determined [2]. Figure 2.1, shows the frame of a manipulator with a single three axis accelerometer attached to the end effector. Although this method seems to be much more beneficial than using an accelerometer mounted to every two joints, similar to what this project proposes, it is crucial that the arm moves as little as possible during calibration. Since the WMRA device is designed to be used with patients, it is paramount that the user’s safety is never jeopardized by erroneous movements. This method [2] could be useful if a rough estimate of the arm’s pose is known. The method proposed in [2] only needs to move each joint twice to successfully calibrate the respective joint. When using this method on a WMRA, two small joint movements may be feasible if the joint values are roughly known or an alternate collision detection system predict that no collisions will occur if the arm moves.
Another method to calibrate the robotic system is by using vision-based calibration. One proposed vision-based calibration method is to mount a checkered board pattern to a specific point on the arm’s end-effector. By using a two dimensional camera mounted to a predetermined location, the camera can detect the position and orientation of the grid pattern [9]. Since the camera frame and the base frame of the arm need to be known before calibration, this method is not feasible in this project. This is because the WMRA’s base frame is unknown relative to the camera frame. This method can work only if the robotic arm is not redundant, and a single solution for inverse kinematics exists. In our case, WMRA is a redundant robot, and infinitely many solutions to the inverse kinematics exist, which makes this method inapplicable.

Another method of vision based calibration is to use multiple cameras mounted on the end-effector of the robotic arm [10]. These cameras attempt to detect checkerboard patterns mounted around the robots environment. This form of calibration is used as an environment based calibration system. In other words, the arm’s exact joint positions are unknown until the end-effector enters the work environment. This method also does not work on redundant manipulators, such as the WMRA being considered in this work.
A stereo vision, or three dimensional camera, can also be used to determine the kinematic pose of a manipulator arm. The WMRA used for this project has an on-board three dimensional camera (Microsoft Kinect) that is used for object recognition and detection [11], environment mapping, robot position calibration [12], and kinematic frame detection of the manipulator arm [13].

Using the method proposed in [13], it is shown that a kinematic model can be constructed using a depth camera. Essentially, the method maps the depth values of the end-effector in multiple different arm configurations. Using this set of end-effector positions, the kinematic model of the robotic arm can be deduced. Again, the drawback of using this camera system to calibrate the arm is that the manipulator arm would be required to move, which is not an option in this project. This method also does not work on redundant manipulators, such as the WMRA considered in this work.

Another vision based calibration technique used previously for this WMRA is the use of a Vicon motion analysis system consisting of multiple high resolution cameras mounted around the robot’s environment [3]. This system uses twelve separate cameras simultaneously to locate small reflector markers mounted around the arm, as shown in Figure 2.2. By using this method of calibration, the position and orientation of the WMRA robotic system can be determined. Although this method solves the calibration problem, it is not feasible when used in the field or outside that environment.

One of the pose detection methods used in [3] and [14] is closely related to this work. Instead of mounting accelerometers after each pair of joints, IMU devices are mounted instead. Chapter 3 will discuss the differences between an IMU device and an accelerometer, but essentially the IMU uses three different measurement devices simultaneously, one of which is a three axis accelerometer. This allows it to detect the exact pitch, roll, and yaw values of the device. The accelerometer is only capable of detecting the pitch and roll values from a three axis accelerometer data.

Another method incorporates an IMU device mounted on the head of a humanoid robot near the on-board cameras [15]. The IMU detects the orientation of the cameras. By detecting the position of point features located on the robotic arm relative to the on-board cameras, the robotic system can be calibrated as a self contained system. Using this calibration method, the robotic arm can be calibrated by making small arm movements and keeping track of the point feature locations.

A similar method of calibration for legged robots also uses an IMU device for orientation sensing [16]. In their work, the robotic system uses a sensor fusion technique that generates an estimate
of the current pose of the legged robot. As the robot moves, the estimate is kept track of and continuously updated. Their sensor fusion technique allows for other sensor devices to be included into the calibration of the robot.

2.2 Key Points

There are a few key points to this project that limit how the robot can be calibrated. One of the main constraints is that the arm must remain stationary while being calibrated. This constraint greatly limits the methods of calibration available. Since, under normal operation, the WMRA will be used with live subjects, the robot must remain stationary during its calibration stage or risk serious injury.

Another constraint is the lack of absolute encoders mounted on the manipulator arm joints. The current joints of the manipulator arm are equipped with dual channel encoders, which allow the user to determine the relative position and direction of spin. Since the arm can be backdriven while powered down, it would be unwise to depend on the last known joint position. With absolute encoders mounted after a gear system, the motor position is known. For this project, absolute
encoders are not a feasible solution since each motor has a gearing system attached after the encoder device, this allows for multiple origin positions on a single joint revolution [5].

The WMRA is a mobile platform, therefore, it is not feasible to have a motion analysis system as a calibration method. The three-dimensional camera mounted above the WMRA (attached to the WMRA) is a possible alternate calibration method for the WMRA system. The manipulator arm can be calibrated by placing colored markers on specific locations on each of the joints [13][17]. The WMRA mobile base can be calibrated by having a known three-dimensional environment map to perform SLAM with [12].

2.3 Summary

It is shown that there are a multitude of different calibration techniques. Each technique has its own benefits and drawbacks. The absolute encoder method is one of the more promising related methods. The most closely related calibration technique is to mount an accelerometer to the robotic arm’s end-effector [2], but this requires arm movement for the calibration to work. Another closely related method of calibration is using IMU devices instead of accelerometers as in [3]. In following chapters, it will be shown how the configuration of the arm has a great effect on the results. Therefore, the calibration method proposed in this project may be more suited as rough estimate. Other methods that acquire more accurate joint estimations but that require joint movement can be used after a rough estimate has been determined.
CHAPTER 3

DEVICE INPUT/OUTPUT

This work incorporates multiple types of embedded devices that communicate with each other using various modes of serial communication. These modes are SPI, I2C and Ethernet.

1. SPI stands for Serial Peripheral Interface. Commonly, it is used to communicate data between two devices, 8-bits at a time. One device typically initiates the transfer and generates the clock signal [18]. In this project, the Raspberry Pi acts as the master device and initiates all communications. There are multiple configurations for SPI, these are based on a predetermined clock phase and clock polarity. For this project, all embedded devices use SPI clock mode 3. That is, data goes on the bus when the input clock is high and the clock will idle with the voltage wire high between data transfers.

2. I2C stands for Inter-Integrated Circuit Bus. This communication is typically used internal to the device, but for the purpose of this project, it will be used to communicate with three of the external embedded systems.

3. Ethernet - The Raspberry Pi is capable of communicating with other computers by using a 10/100 wired Ethernet connection. A UDP server protocol was setup on the Raspberry Pi to send all raw and computed data. It is also used to communicate with the WMRA motor controller when arm movement is necessary.

The reason for using two types of communication protocols is due to limitations in both the accelerometers and the Raspberry Pi. This will be discussed in more detail later in this chapter.

3.1 Raspberry Pi

The Raspberry Pi device is capable of supporting multiple Input/Output devices. The input devices used for this work are the MPU-9150 IMU device and multiple ADXL-345 three axis
accelerometers. Two of the accelerometers communicate using the SPI communication protocol while the other two use the I2C protocol. The Raspberry Pi communicates with these devices at a 100kHz clock frequency. Since the arm is stationary when calibration is in progress, the selected communication speed is more than enough for this project.

The Raspberry Pi is mounted to the base of the WMRA. It is capable of connecting and communicating with the WMRA motor controller. The Raspberry Pi is used to calibrate the robotic system without external assistance. A Linux type operating system, Raspbian, is used for this project. Most Linux standard libraries and many third party libraries can be used on this device. The library, WiringPi, is used for I2C communication with two of the accelerometers. The library, BCM-2835, is used for SPI communication with the other two accelerometers [19]. Although all of the computations are done on the Raspberry Pi, the data can still be retrieved by connecting to the UDP server which is setup on the Raspberry Pi.

3.2 Inertial Measurement Unit

The Inertial Measurement Unit (IMU) used in this project is the InvenSense MPU-9150 [7]. The Raspberry Pi communicates with this device using the I2C serial protocol. It uses the same clock and data bus as the I2C accelerometers but with a different device ID. The four wires the IMU uses are Voltage (Vcc), Ground (GND), Device In/Out (SDA), and Clock (SCL). The IMU is set to read linear acceleration forces ranging between $\pm 2g$. This device outputs data from three separate inertial data devices. The three devices are: three axis accelerometer, a three axis gyroscope, and a three axis magnetometer. Using a fusion of these devices, the orientation and cardinal direction can be determined for the base frame of the WMRA.

3.3 Accelerometers

The accelerometers used in this study [8] are capable of communicating in both I2C and SPI serial protocols. Both types of communication are used in this project due to certain device constraints that will be discussed later. The two modes of communication are:

1. SPI - This work is using the 4-Wire SPI communication configuration, as seen in Figure 3.1. Although the maximum SPI clock speed is 5MHz, we will only be reading at 1MHz
since a fast update speed is not required for this work. Two of the four accelerometers will be using SPI.

![Figure 3.1: ADXL 345 accelerometer using SPI 4-wire communication](image)

These SPI accelerometers require more wires than their I2C counterparts and will be positioned closer to the base of the arm to limit the number of wires that need to be placed on the arm. The six wires of the SPI accelerometers are Voltage (Vcc), Ground (GND), Chip Select (CS), Device Out (SDO), Device In (SDA), and Clock (SCL).

2. I2C - The accelerometers used for this project are also capable of using I2C communication. This is achieved by tying the chip select pin to Vcc. This will disable the SPI capabilities and allow for the chip to be used in I2C mode. This wiring setup is displayed in Figure 3.2. Since the chip select is not used to activate the device, the individual accelerometer data is accessed by using a device address that is specific to each accelerometer. The accelerometers used in this work have a main address and an alternate address hard coded into the chip. There are only two I2C accelerometers used in this project, so both the main and alternate IDs will be used. This is one reason why four accelerometers using I2C communication is not feasible for this project.

By using an addressing scheme (I2C) instead of a chip select (SPI), there are only four wires required to communicate with the accelerometers using I2C communication. These four wires are Voltage (Vcc), Ground (GND), Device In/Out (SDA), and Clock (SCL).

The wiring diagram in Figure 3.3 shows how the accelerometers and inertial measurement unit are connected to the Raspberry Pi. The Raspberry Pi uses general purpose input/output pins to power and communicate with the sensing devices.
Figure 3.2: ADXL 345 accelerometer using I2C communication

Figure 3.3: Wiring diagram for the embedded devices
CHAPTER 4

WMRA BASE FRAME CALIBRATION

The base frame of the kinematic chain for the WMRA is found using an Inertial Measurement Unit (IMU). This device, when properly calibrated, will be able to determine the orientation of the WMRA base frame for the pitch and roll axes.

Figure 4.1: WMRA base frame [1]
The cardinal direction, yaw, is determined by using the IMU’s three dimensional magnetometer. The IMU coordinate frame is also mapped to the wheelchair coordinate frame to determine the current wheelchair positioning.

The IMU is mounted at the origin of the base of the manipulator arm to compute the base frame. The base of the manipulator is located at, $\sum_{\text{ArmBase}}$, from Figure 4.1. The pitch and roll angles, in radians, represent the rotation around the X-axis and Y-axis of the IMU accelerometer. The yaw measurement, around the Z-axis, represents the rotational angle with the X-axis of the gravity frame as cardinal north. Figure 4.1 shows the X-axis of the WMRA base frame is pointing forwards.

4.1 Orientation

The Roll and Pitch of the base frame can be determined using the IMU’s three dimensional accelerometers. The yaw cannot be determined by using the accelerometers alone due to the gimbal lock problem [20]. At the beginning of this work, it was assumed that the robotic system would always be calibrated on a level surface. This may not always be the case when used in the field. Therefore, the base frame must be computed before any of the manipulator arm joints can be calibrated. It was later discovered that the WMRA itself is not completely level across the wheelchair base, so the placement of the IMU needed to be mounted as close to the arms base frame as possible. The pitch and roll angles found from the IMU are used in the rotation matrix below.

$$R^0_W = \begin{bmatrix}
\cos(\alpha) & 0 & -\sin(\alpha) \\
\sin(\alpha) \cdot \sin(\psi) & \cos(\psi) & \cos(\alpha) \cdot \sin(\psi) \\
\sin(\alpha) \cdot \cos(\psi) & -\sin(\psi) & \cos(\alpha) \cdot \cos(\psi)
\end{bmatrix}$$ (4.1)

$R^0_W$ is the rotation matrix representing the pitch($\alpha$) and roll($\psi$) rotations of the accelerometer around its X-axis and Y-axis respectively. This $R^0_W$ rotation matrix will be used as the base frame of the manipulator arm. This rotation matrix is setup similar to the other accelerometers.
4.2 Cardinal Direction

The yaw, or cardinal direction, is used to show the rotation around the Z-axis of the wheelchair frame. Although this information is not necessary for the calibration of the manipulator arm, it does provide useful information for other localization techniques. The issue when determining the yaw value is that the magnetometer on the IMU device needs to be pre-calibrated for the area it is working in. There is also an issue of metals and other electromagnetic material interfering with the results. Since the three axis magnetometer is the device used to determine the cardinal direction, any magnetic fields near the magnetometer will cause noise in the data when detecting magnetic north.
In this chapter, the proposed calibration technique that utilizes multiple three dimensional accelerometers to determine pairs of kinematic joint angles is presented in detail. The manipulator used for testing consists of seven rotational joints to create a seven degree of freedom robotic arm system. As discussed in Chapter 2, this manipulator is capable of determining its relative position from the time the system is given power and onward. As mentioned in previous chapters, the robotic system used in this work is not equipped with absolute encoders. Due to the arm’s movement while powered down, the current position of each joint in the arm is considered unknown.

For this project, the accelerometers are required to be placed no more than one per two rotational joints. The pairs of rotation matrices that will be computed within this project are $R_{0}^{2}$, $R_{2}^{4}$, $R_{4}^{6}$ and the single joint value in $R_{6}^{7}$. The rotational matrices below are used later in this chapter when determining the joint values for joints one to seven. For each accelerometer, a pair of joints is calculated, therefor, the rotational matrices $R_{0}^{2}$, $R_{2}^{4}$, $R_{4}^{6}$ and $R_{6}^{7}$ are needed to determine the joint values. Equations 5.1 and 5.2 are used to determine the values of joints one and two from the first accelerometer’s data, and are discussed in Section 5.1.

$$R_{0}^{1} = \begin{bmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) \\ -\sin(\theta_1) & 0 & -\cos(\theta_1) \\ 0 & 1 & 0 \end{bmatrix} ; \quad R_{1}^{2} = \begin{bmatrix} \cos(\theta_2) & 0 & \sin(\theta_2) \\ -\sin(\theta_2) & 0 & \cos(\theta_2) \\ 0 & -1 & 0 \end{bmatrix}$$ (5.1)

$$R_{0}^{2} = R_{1}^{2} \ast R_{0}^{1} = \begin{bmatrix} \cos(\theta_1) \ast \cos(\theta_2) & \sin(\theta_2) & -\cos(\theta_2) \ast \sin(\theta_1) \\ -\cos(\theta_1) \ast \sin(\theta_2) & \cos(\theta_2) & \sin(\theta_1) \ast \sin(\theta_2) \\ \sin(\theta_1) & 0 & \cos(\theta_1) \end{bmatrix}$$ (5.2)

Equations 5.3 and 5.4 are used in the same manner when determining joints three and four, and are discussed in Section 5.2.
\[ R_2^3 = \begin{bmatrix} \cos(\theta_3) & 0 & -\sin(\theta_3) \\ -\sin(\theta_3) & 0 & -\cos(\theta_3) \\ 0 & 1 & 0 \end{bmatrix}; \quad R_3^4 = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) \\ -\sin(\theta_4) & 0 & \cos(\theta_4) \\ 0 & -1 & 0 \end{bmatrix} \quad (5.3) \]

\[ R_2^4 = R_3^4 \times R_2^3 = \begin{bmatrix} \cos(\theta_3) \times \cos(\theta_4) & \sin(\theta_4) & -\cos(\theta_4) \times \sin(\theta_3) \\ -\cos(\theta_3) \times \sin(\theta_4) & \cos(\theta_4) & \sin(\theta_3) \times \sin(\theta_4) \\ \sin(\theta_3) & 0 & \cos(\theta_4) \end{bmatrix} \quad (5.4) \]

Equations 5.5 and 5.6 are used to determine joints 5 and 6 from the third accelerometer’s data, and are discussed in Section 5.3.

\[ R_4^5 = \begin{bmatrix} \cos(\theta_5) & 0 & -\sin(\theta_5) \\ -\sin(\theta_5) & 0 & -\cos(\theta_5) \\ 0 & 1 & 0 \end{bmatrix}; \quad R_5^6 = \begin{bmatrix} \cos(\theta_6) & 0 & \sin(\theta_6) \\ -\sin(\theta_6) & 0 & \cos(\theta_6) \\ 0 & -1 & 0 \end{bmatrix} \quad (5.5) \]

\[ R_4^6 = R_5^6 \times R_4^5 = \begin{bmatrix} \cos(\theta_5) \times \cos(\theta_6) & \sin(\theta_6) & -\cos(\theta_6) \times \sin(\theta_5) \\ -\cos(\theta_5) \times \sin(\theta_6) & \cos(\theta_6) & \sin(\theta_5) \times \sin(\theta_6) \\ \sin(\theta_5) & 0 & \cos(\theta_5) \end{bmatrix} \quad (5.6) \]

Equation 5.7 is used to determine joint seven using the fourth accelerometer’s data, and is discussed in Section 5.4.

\[ R_6^7 = \begin{bmatrix} \cos(\theta_7) & 0 & -\sin(\theta_7) \\ -\sin(\theta_7) & 0 & -\cos(\theta_7) \\ 0 & 1 & 0 \end{bmatrix} \quad (5.7) \]

Table 5.1 shows the WMRA's DenavitHartenberg parameters that are used in this work to determine the rotation matrices for Equations 5.2, 5.4, 5.6, and 5.7. These DH-parameters are going to be different for other types of robotic manipulators, and the rotation matrices generated should reflect the changes. The DH-parameters are first used to find the transformation matrices of the robotic arm. The rotation matrix portion of a transformation matrix is the 3 x 3 matrix starting from the top left portion of the selected transformation matrix. These matrices are shown in Equations 5.1, 5.3, 5.5, and 5.7. The rotation matrices used in Equations 5.2, 5.4, 5.6, and 5.7 are taken from the transformation matrices found from the robot’s DH-parameters.
When solving for each pair of joints, there are three equations with two unknowns. To solve for the unknown joint angles numerically, the Levenberg-Marquardt method is used \([21]\). The Levenberg-Marquardt method is also known as the Damped Least-Squares method which attempts to find the local minimum of a given problem. It is considered the standard technique to use when solving nonlinear least squares problems. This method is a combination of two well known minimization methods: the gradient descent method and the Gauss-Newton method \([21]\). This method converges quickly when an initial point close to the actual joint value is given. The initial point given at the start of the program is the arm’s natural starting position with the assumption that the arm will be near that position when the system is started. It should be noted that a solution can still be determined if the initial position given is not near the current joint location, the solution will just take longer to determine. For this setup, it usually takes six to ten iterations of the loop to determine a local minimum.

Figure 5.1 shows the arm of the robot used in the work, and the frames of each joint are also located in the Figure. The red squares show the placement of accelerometers one through four on the manipulator. Although intuitively it seems that with a three axis accelerometer, the spacing should be one for every three rotational joints. Since the accelerometer data gives a unit vector with a fixed length of 1g, the three axis accelerometer only has two degrees of freedom. When detecting the manipulator calibration values, an error occurs as the axes of rotation for any joint aligns with the gravity vector. When this parallel configuration occurs, the system is considered to be in a gimbal lock configuration \([20]\). This will be discussed further in Chapter 6.
5.1 Joints 1 and 2

To solve for joints one and two, a kinematic Equation must be formulated. First, a kinematic chain of rotation matrices is constructed to solve for the unit vector of gravity, as seen in Equation 5.8. The vector shown at the end of the kinematic Equation allows only the Z-axis of the computed rotation matrix to be used. This generates a 3 x 1 unit vector which will match the known accelerometer readings at any given joint positions.

$$G = R_2^A \cdot R_0^2 \cdot R_W^0 \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$ (5.8)
From Equation 5.8, $R_{2A1}^{A1}$ is the rotation matrix that represents how accelerometer one is mounted to the link past joint 2 of the manipulator. This rotation matrix is known and determined before hand and is based on the orientation of the accelerometer mounted on the manipulator frame. For this project, $R_{2A1}^{A1}$ is shown in Equation 5.9, and represents the rotation from frame 2 to frame A1 shown in Figure 5.2.

$$R_{2A1}^{A1} = \begin{bmatrix}
0 & -1 & 0 \\
0 & 0 & 1 \\
-1 & 0 & 0
\end{bmatrix}$$  \hspace{1cm} (5.9)

$R_{0W}^2$ is the rotation matrix that holds the two unknowns to be solved for, $\theta_1$ and $\theta_2$, and is shown in Equation 5.2.

$R_{0W}^0$ represents the orientation of the base frame of the WMRA relative to the ground and is used as the origin for the manipulator. $R_{0W}^0$ is computed using the calculated pitch and roll values from the IMU device mounted to the base of the WMRA. The $R_{0W}^0$ matrix is shown in Figure 4.1, where the pitch and roll are represented as $\alpha$ and $\psi$ respectively.

$G$ is a unit vector in the direction of gravity. For this project, $G$ represents the accelerometer data converted to unit vector form. To do this, the raw accelerometer data must be processed. This procedure starts by receiving raw accelerometer data via the SPI connection to the Raspberry Pi. The accelerometer outputs magnitudes of force on each of the three axes. This data is sent in its raw form as an integer that ranges from 0 to 1023 based on the range selected for this project.
There are three separate 10-bit integers for each axis of the accelerometer. These values are first normalized to create a unit vector of the gravitational vector.

This gravity unit vector is denoted as, \( G = [G_x, G_y, G_z] \). To normalize the raw accelerometer data \( (A_{1x}, A_{1y}, A_{1z}) \), first the length of the vector, \(|a|\), is computed using Equation 5.11. Divide each of the accelerometers X, Y, and Z components by the vector length, this is shown in Equation 5.11 [22].

\[
|a| = \sqrt{A_{1x}^2 + A_{1y}^2 + A_{1z}^2} \quad (5.10)
\]

\[
G_x = \frac{A_{1x}}{|a|} \quad G_y = \frac{A_{1y}}{|a|} \quad G_z = \frac{A_{1z}}{|a|} \quad (5.11)
\]

The rotation matrix in Equation 5.2 holds the two unknown joint angles that are to be solved. The \( R_1^2 \) and \( R_0^1 \) matrices for this project are shown in Equation 5.1. To solve the \( \theta_1 \) and \( \theta_2 \) joint angles from \( R_0^2 \) rotation matrix, the Levenberg-Marquardt algorithm is used. The Levenberg-Marquardt algorithm, also known as the damped least-squares method, is used to solve this non-linear problem. For the first pair of joints, there are two variables to solve for \( (\theta_1 \text{ and } \theta_2) \) and three equations from equating the elements of the vectors in Equation 5.8. The fsolve function in MATLAB was used to solve these equations for \( \theta_1 \) and \( \theta_2 \) using the Levenberg-Marquardt algorithm.

5.2 Joints 3 and 4

The calibration of joints three and four will be determined by using the computed values of joints one and two along with the second accelerometers data. Therefore, it is necessary to compute the rotation matrix for \( R_0^2 \) before continuing to solve the \( R_4^2 \) rotation matrix. Once the \( R_0^2 \) matrix is solved, the process of computing \( R_4^2 \) is similar to solving for the \( R_0^2 \) matrix. The kinematic chain of rotation matrices used to solve for the unit vector of gravity, can be seen in Equation 5.12. The vector shown at the end of the kinematic Equation allows only the Z-axis of the computed rotation matrix to be used. This generates a \( 3 \times 1 \) unit vector which will match the known accelerometer readings at any given joint positions.
From Equation 5.12, \( R_{4}^{A_2} \) is the rotation matrix that represents how accelerometer two is mounted to the link past joint 4 of the manipulator. This rotation matrix is known and determined beforehand and is based on the orientation of the accelerometer mounted to the manipulator frame. For this project, \( R_{4}^{A_2} \) is shown in Equation 5.13, and represents the rotation from frame 4 to frame A_2 shown in Figure 5.3.

![Figure 5.3: Accelerometer two frame to joint four frame](image)

Although Equation 5.13 is similar to Equation 5.9, there are few key differences. The rotation matrix from frame 2 to frame 4, \( R_{4}^{2} \), was added to the Equation. Also, the value for \( R_{4}^{A_2} \) is different from Equation 5.9 and is dependent on the output of accelerometer two instead of accelerometer one. When determining the values of joints one and two, the rotation matrix \( R_{2}^{3} \) held the two unknowns \( \theta_1 \) and \( \theta_2 \). Similarly, the joint three and four unknowns can be found from the \( \theta_3 \) and \( \theta_4 \) values in the \( R_{2}^{4} \) rotation matrix.
\( R^4_2 \) is the rotation matrix that holds the two unknowns to be solved for, \( \theta_3 \) and \( \theta_4 \), and is shown in Figure 5.4.

\( R^0_W \) represents the orientation of the base frame of the WMRA relative to the ground and is used as the origin for the manipulator. \( R^0_W \) is computed using the calculated pitch and roll values from the IMU device mounted to the base of the WMRA. The \( R^0_W \) matrix is shown in Figure 4.1, where the pitch and roll are represented as \( \alpha \) and \( \psi \) respectively.

\( G \) is a unit vector in the direction of gravity. For kinematic chain, \( G \) represents the second accelerometer data converted to unit vector form. To do this, the same method used in Section 5.1 is used. Equations 5.10 and 5.11 are used to create the unit vector for accelerometer two.

The rotation matrix in Equation 5.4 holds the two unknown joint angles that are to be solved. The \( R^4_3 \) and \( R^3_2 \) matrices for this project are shown in Equation 5.3. To solve the \( \theta_3 \) and \( \theta_4 \) joint angles from \( R^4_2 \) rotation matrix, the Levenberg-Marquardt algorithm is used. For the second pair of joints, there are two variables to solve from three equations. The fsolve function in MATLAB was used to solve these equations for \( \theta_3 \) and \( \theta_4 \) using the Levenberg-Marquardt algorithm. Once \( R^4_2 \) rotation matrix is computed, it is then combined with \( R^2_0 \) to create \( R^4_0 \) rotation matrix that will be used when determining joints five and six.

5.3 Joints 5 and 6

The calibration of joints five and six will be determined by using the computed values of joints one through four along with the third accelerometers data. Therefore, it is necessary to compute the rotation matrix for \( R^5_0 \) and \( R^4_3 \) before continuing to solve the \( R^6_4 \) rotation matrix. Once the \( R^6_4 \) matrix is solved, the process of computing \( R^6_0 \) is similar to solving for the \( R^4_2 \) matrix. The kinematic chain of rotation matrices used to solve for the unit vector of gravity can be seen in Equation 5.14. The vector shown at the end of the kinematic Equation allows only the Z-axis of the computed rotation matrix to be used. This generates a 3 x 1 unit vector which will match the known accelerometer readings at any given joint positions.

\[
G = R^A_6 \ast R^6_3 \ast R^4_2 \ast R^2_0 \ast R^0_W \ast \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
\]  
(5.14)
From Equation 5.14, $R_{6}^{A_3}$ is the rotation matrix that represents how accelerometer three is mounted to the link past joint 6 of the manipulator. This rotation matrix is known and determined before hand and is based on the orientation of the accelerometer mounted on the manipulator frame as shown in Equation 5.15.

$$R_{6}^{A_3} = \begin{bmatrix}
0 & -1 & 0 \\
0 & 0 & 1 \\
-1 & 0 & 0
\end{bmatrix}$$  \hspace{1cm} (5.15)

Although Equation 5.15 is similar to Equation 5.13, there are few key differences. The rotation matrix from frame 4 to frame 6, $R_{6}^{A_3}$, was added to the Equation. Also, the value for $R_{6}^{A_3}$ is different from Equation 5.13 and is dependent on the output of accelerometer three instead of accelerometer two. When determining the values of joints three and four, the rotation matrix $R_{4}^{A_3}$ held the two unknowns $\theta_3$ and $\theta_4$. Similarly, the joint five and six unknowns can be found from the $\theta_5$ and $\theta_6$ values in the $R_{6}^{A_3}$ rotation matrix.

$R_{6}^{A_3}$ is the rotation matrix that holds the two unknowns to be solved for, $\theta_5$ and $\theta_6$, and is illustrated in Figure 5.6.

$R_{W}^{0}$ represents the orientation of the base frame of the WMRA relative to ground and is used as the origin for the manipulator. $R_{W}^{0}$ is computed using the calculated pitch and roll values from
the IMU device mounted to the base of the WMRA. The $R^0_{W}$ matrix is shown in Figure 4.1, where the pitch and roll are represented as $\alpha$ and $\psi$ respectively.

$G$ is a unit vector in the direction of gravity. For this kinematic chain, $G$ represents the third accelerometer data converted to unit vector form. To do this, the same method used in Section 5.1 is used. Equations 5.10 and 5.11 are used to create the unit vector for accelerometer three.

The rotation matrix in Equation 5.6 holds the two unknown joint angles that are to be solved. The $R^6_5$ and $R^5_4$ matrices for this project are shown in Equation 5.5. To solve the $\theta_5$ and $\theta_6$ joint angles from $R^6_4$ rotation matrix, the Levenberg-Marquardt algorithm is used. For the third pair of joints, there are two variables to solve for from three equations. The fsolve function in MATLAB was used to solve these equations for $\theta_5$ and $\theta_6$ using the Levenberg-Marquardt algorithm. Once $R^6_4$ rotation matrix is computed, it is then combined with $R^4_0$ to create $R^6_0$ rotation matrix that will be used when determining joint seven.

5.4 Joint 7

The calibration of joint seven will be determined by using the computed values of joints one through six along with the fourth accelerometers data. Therefore, it is necessary to compute the rotation matrix for $R^2_0$, $R^4_2$ and $R^6_0$ before continuing to solve the $R^7_6$ rotation matrix. Once the $R^6_0$ matrix is solved, the process of computing $R^7_6$ is similar to solving for the $R^6_0$ matrix. The kinematic chain of rotation matrices used to solve for the unit vector of gravity, can be seen in Equation 5.16. The vector shown at the end of the kinematic Equation allows only the Z-axis of the computed rotation matrix to be used. This generates a 3 x 1 unit vector which will match the known accelerometer readings at any given joint positions.

\[
G = R^{A_4}_7 \ast R^7_6 \ast R^6_0 \ast R^4_2 \ast R^2_0 \ast R^0_W \ast \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{5.16}
\]

From Equation 5.16, $R^{A_4}_7$ is the rotation matrix that represents how accelerometer four is mounted to the link past joint 7 of the manipulator. This rotation matrix is known and determined before hand and is based on the orientation of the accelerometer mounted on the manipulator frame as is shown in Equation 5.17.
Although Equation 5.17 is similar to Equation 5.15, there are few key differences. The rotation matrix from frame 6 to frame 7, $R_{76}$, was added to the Equation. Also, the value for $R_{74}$ is different from Equation 5.15 and is dependent on the output of accelerometer four instead of accelerometer three. When determining the values of joints five and six, the rotation matrix $R_{64}$ held the two unknowns $\theta_5$ and $\theta_6$. Similarly, the joint seven unknown can be found from the $\theta_7$ value in the $R_{76}$ rotation matrix.

$R_{76}$ is the rotation matrix that holds the single unknown to be solved for, $\theta_7$, and is illustrated in Figure 5.7.

$R_{0W}$ represents the orientation of the base frame of the WMRA relative to the ground and is used as the origin for the manipulator. $R_{0W}$ is computed using the calculated pitch and roll values from the IMU device mounted to the base of the WMRA. The $R_{0W}$ matrix is shown in Figure 4.1, where the pitch and roll are represented as $\alpha$ and $\psi$ respectively.

$G$ is a unit vector in the direction of gravity. For this kinematic chain, $G$ represents the fourth accelerometer data converted to unit vector form. To do this, the same method to create a unit vector in Section 5.1 is used. Equations 5.10 and 5.11 are used to create the unit vector for accelerometer four.
The rotation matrix, $R_{6}^{7}$, in Equation 5.7 holds the unknown joint angle that is to be solved. To solve the $\theta_{7}$ joint angle from the $R_{6}^{7}$ rotation matrix, the Levenberg-Marquardt algorithm is used. For the last joint, there is one variable to solve for from three equations. The `fsolve` function in MATLAB was used to solve these equations for $\theta_{7}$ using the Levenberg-Marquardt algorithm. This is the last rotational joint in the robotic arm being calibrated for this work. The robotic system should now be completely calibrated as long as the arm is in a configuration that adds minimal error. This error caused when certain joint configurations are reached is discussed further in the next chapter.
CHAPTER 6

MANIPULATOR ARM CONFIGURATION ERRORS

In the previous chapters, a method was discussed to calibrate a manipulator with three axis accelerometers. The calculations discussed work accurately in a perfect environment when sensors of high accuracy are used. With the accelerometers and dual channel encoders used in this work, an error occurs that ranges from 0% to 100% depending on the current configuration of the arm. The error in the computations directly depends on the angle of the z-axis of the accelerometer frame relative to the x-y plane of the gravity frame. Once an axis of rotation aligns with the vector of gravity the estimated joint value becomes indeterminate due to a divide by zero problem that occurs when solving.

There are two different errors that can greatly effect the output of this calibration method. The error incurred when the accelerometer’s Z-axis is close to horizontal is one of the main problems. In this work, we refer to this error as the "pitch and roll error."

The more joints the manipulator arm has, the more accelerometers are needed to solve the calibration problem. Each rotation matrix has its own pitch and roll error. This error is cumulative since each accelerometer uses the preceding accelerometer’s value as its base frame. This causes the joints towards the end of the arm to have larger error when determining their joint values.

6.1 Pitch and Roll Error

As mentioned previously, the error when detecting the accelerometer’s rotation matrix is a major problem when using this calibration method. With the accelerometers available today, the accuracy is not high enough to eliminate this error, and specific configurations should be avoided. The amount of error can be computed as a function of roll and pitch angles, which are $\psi$ and $\alpha$ respectively in Equation 6.2. Equation 6.1 solves for the current error in pitch and roll that the accelerometer is experiencing [4].
\[ \Delta \epsilon = \tan^{-1} \left[ \frac{G_y (G_z - \sqrt{G_z^2 + G_x^2})}{G_y^2 + G_z \sqrt{G_z^2 + G_x^2}} \right] \] (6.1)

\[ \tan^{-1} \left[ \frac{\sin(\psi)(\cos(\alpha) \cos(\psi) - \sqrt{\cos^2 \alpha \cos^2 \psi + \sin^2 \alpha})}{\cos \alpha \sin^2 \psi + \cos \psi \sqrt{\cos^2 \alpha \cos^2 \psi + \sin^2 \alpha}} \right] \] (6.2)

The pitch and roll errors increase as the Z-axis of the accelerometer frame gets closer to the x-y plane of the gravity frame. Figure 6.1 shows the change in error as either the pitch or roll values start to move from the zero angle. Using the normalized accelerometer data, the error present when computing the rotation matrix is given using Equation 6.1. Figure 6.1 shows the amount of error generated when normalized accelerometer data is used in Equation 6.1.

![Figure 6.1: Pitch and roll error range [4].](image)

The best case for any accelerometer is when the Z-axis of the accelerometer frame is aligned with the Z-axis of the gravity frame. In this case, pitch and roll angles would both be zero and there would be minimal error that would effect the results for that accelerometer.

On the other end of the scale, as either pitch or roll approaches 90°, the error rises from 0% to 100%. This can cause a problem with this calibration method and may require small movements of the arm to move the Z-axis out of the undesired alignment.
6.2 Cumulative Error

It should be apparent that since previous accelerometer frames are used as the base frames for accelerometers further down the manipulator, the pitch and roll errors would also accumulate. This error for later joints gets worse as more joints, thus accelerometers, are added to the arm.

Once an accelerometer is near an incorrect alignment, the accelerometers afterwards have an incorrect joint frame to reference from. A serious problem occurs if this improper alignment happens with the first pair of joints. Since none of the arm’s joints are known, any movement could cause dangerous collisions.

6.3 Configuration Error

As mentioned above, with the proper configurations, the joint values computed by the accelerometers can have small amounts of error. But with each extra accelerometer included for calibration, more error can accumulate.

With the accelerometers mounted in their current position, an example of configuration error occurs when joints one and two are at ninety degrees and joints three and four are at zero degrees. In this arm configuration, accelerometer one is in the perfect alignment to detect the values of joints one and two with minimal error. Since the Z-axis of the first accelerometer, which is also the axis of rotation for joint three, is now parallel with the gravity vector, joint three can’t be properly computed.

To correct this problem and to reduce error in the computed output, individual joint movement may be necessary. Joint movement is not preferable and should not be done with a joint that is not at least roughly calibrated.
CHAPTER 7

EXPERIMENTAL RESULTS

7.1 Testing Procedure

The testing procedure consists of pre-calibrating the WMRA, including the base orientation as well as the joint values of the manipulator arm. The WMRA will then be placed into multiple poses and data will be collected from each accelerometer. The IMU device will be tested separately at first to determine that the correct base frame is generated for multiple orientations and changes in cardinal direction. The accelerometers will initially be tested on a level surface. Once these tests are complete, tests will then be conducted with the WMRA wheelchair tilted at varying orientations, within reason. For instance, the most inclination the WMRA will be tested at will be no more than the inclination of a standard wheelchair ramp.

The testing of each pair of joints in the WMRA’s kinematic chain will be done from the base of the arm to the end-effector. For example, the joint values generated by accelerometer one will be tested against the known joint angles for joints one and two. Once accelerometer one is tested, accelerometer two will then be tested. This will continue until all computed joint values have been tested against their respective known joint values.

It is expected that there will be cumulative error in the noise from each accelerometer. Therefore, it is expected that joints one and two will be the most accurate, while joint seven of the manipulator arm will have the most error. This will be due to the error generated from the first accelerometer being added to the error of the next accelerometer and so on.

7.2 Data and Results

The first testing procedure will involve estimating the WMRA arm base frame using the inertial measurement unit mounted at the origin of the manipulator. As mentioned in Chapter 4, the IMU
device is able to determine not just pitch and roll, but also the yaw value as well. The yaw value will help to determine the cardinal direction of the WMRA in its environment. The pitch and roll values will be used to determine the orientation of the manipulator origin. To begin, the manipulator is placed in a modified ready position with joint one at 45° instead of 90°. This will allow all joint values to be accurately estimated from the start of the test. The WMRA base will then be tilted along the Y-axis of the Arm Base, shown in Figure 4.1. Figures 7.1 and 7.2 show the estimation of the first two joints when the wheelchair is on an inclined surface. The difference between the equations used to estimate the joint values is in Figure 7.1 and 7.2 is Figure 7.1 shows the equation without using the $R^0_W$ rotation matrix to offset for the inclination while the equation for Figure 7.2 uses the IMU to detect the inclination offset. Notice in Figure 7.2 that even though the origin is tilted, the estimated joint values remain constant at their current values.

![Figure 7.1: Joints 1 and 2 estimations using a 3 x 3 identity matrix as $R^0_W$](image)
The next testing procedure will consist of moving the arm into a pose that will cause joints three and six to align with the vector of gravity. It can be seen in Figures 7.3 and 7.4 that as the axis of revolution aligns with the vector of gravity, the computed joint value become erroneous, eventually becoming indeterminate. For this test, joint one is first positioned so no axes of revolution align with the vector of gravity. Then joint one is moved through alignment with gravity. This can be seen at the 250\textsuperscript{th} data set. After the 500\textsuperscript{th} data set, Figure 7.3 and 7.4 shows how the amount of error in the computed joint value increases as joint one steps closer to alignment with the vector of gravity. When estimating the joint values of the manipulator, alignments near or with the vector of gravity can be detected before joint estimation attempted. This alignment detection can be used to inform the user that the arm is in a non-calibratable position. Since there is an axis of one of the accelerometers that is always parallel with a specific revolute joint, alignments with the gravity vector can be detected by reading the magnitude of acceleration present on the accelerometers parallel axis. As the axis of the accelerometer that is parallel with the axis of revolution starts to align with the vector of gravity, then the magnitude of force detected on that axis will approach
1g. When the axis has a magnitude of 1g, then the axis of revolution of the specific joint is aligned with the gravity vector.

Figure 7.3: Joints 1 through 4, with joint 1 moving through gravity vector alignment

Figure 7.4: Joints 5 through 7, with joint 1 moving through gravity vector alignment
The final test protocol consists of moving the manipulator to different joint configurations and detecting the current joint angles for each joint. Each of the poses used in the test keep all axes of rotation away from alignment with the vector of gravity. Figures 7.5, 7.6, 7.7 and 7.8 show various manipulator poses. It can be observed that the various manipulator poses produce varying ranges of error. As mentioned in the previous chapter, the error seen can be introduced by the different types of error present in this calibration method. It should be noted that the joints closer to the base of the arm tend to be more accurate than the joint estimations further down the kinematic chain. For example, the computed values for joints one and two in Figure 7.5 are very close to the known joint values. While the computed values for joints five and six in Figure 7.7 have more error in the results than the previous joint estimations. This increase in the amount of error is related to the error present in the estimations of prior joints.

![Figure 7.5: Joints 1 and 2 computed values](image)

![Figure 7.6: Joints 3 and 4 computed values](image)

![Figure 7.7: Joints 5 and 6 computed values](image)

![Figure 7.8: Joint 7 computed values](image)
For this calibration method, an error of ±0.1 radians is considered acceptable for joint estimation. For all seven joints, the error amount seen in the readings is within this range while the axes of rotation are angled away from the vector of gravity. With a more accurate calibration method for the mounting of the inertial measurement unit (R0W) and accelerometers (R2A1, R4A2, R6A3, R7A4), the joint estimations may be more accurate. Joints one through four in Figures 7.5 and 7.6 show much better results with less error than joints five and six in Figure 7.7. This is an effect of the cumulative error from the previous joints causing the joint estimations of joints five and six to be partially erroneous. It is good to notice that the joint estimation for joint seven in Figure 7.8 has much less error in the results than the estimations for joints five and six.
CHAPTER 8

CONCLUSION

It is shown that by using multiple three axis accelerometers mounted at specific locations on a robotic manipulator, the absolute values for each joint can be determined. By adding an inertial measurement unit (IMU) to the base frame, the mobile robotic platform can be at almost any incline and still be able to detect the correct position of the arm, as long as none of the axes of revolution for the manipulator align with the vector of gravity. The main limitation of this method is when the manipulator is in certain configurations, where an axis of revolution aligns with the vector of gravity, the estimated joint values becomes indeterminate.

8.1 Overview

This work focuses on the calibration of a 7-DOF robotic manipulator attached to a power wheelchair, by using multiple three axis accelerometers. It is seen early in the project that the resolution of the accelerometers and the error in the physical mounting can cause an accumulation of error that makes this method of calibration relatively inaccurate. This method of calibration should be used as an initial or secondary calibration method to calculate a rough estimate of the joint angles. Although the joints further down the robotic arm have a larger accumulation of error, this rough estimate can be enough to allow for safe calibration using more precise methods of calibration that require arm movement.

As mentioned in previous chapters the order in which the joints of the manipulator are solved is important. The tilt of the base frame of the manipulator arm must be solved before any part of the arm can be calibrated. Figure 8.1 shows the order in which joints one through seven must be solved. Once a joint is solved for and the system isnt aligned with the vector of gravity, then the absolute joint angles can be communicated to the motor control for the manipulator.
8.2 Future Work

As seen in the results, the cumulative error between each accelerometer becomes a serious issue when using this calibration method. One possible way to eliminate this error is to use the accelerometers’ readings as a rough estimate to the true joint angles. By using absolute encoders and this rough estimate, even with a gearing system attached, it may be possible to determine the absolute joint angle of the arm.

Also, some of the related works mentioned in Chapter 2 cannot be safely used on an un-calibrated WMRA, but could be used once a rough estimate of the joint angle in known. For example the proposed calibration method in [2] will not work for calibrating the WMRA because the device does not know the current joint values before moving the manipulator. This issue can be overcome by initially calibrating the robotic manipulator using the method proposed in this thesis, then using the method in [2] to fine tune the computed joint values.

Another similar method to calibrate the robotic manipulator would be to use inertial measurement units instead of the accelerometers used in this work. By using IMU devices a 3rd degree of freedom can be detected per device. This means only one IMU device would need to be mounted after every three joints instead of every two joints.

8.3 Concluding Remarks

It is important to realize that this is not the optimal way to calibrate a robotic system using an accelerometer. This is mainly due to the method giving an indeterminate joint estimation at certain
manipulator poses. With that being said this is a valid method of calibration when movement is not permitted during the initial calibration procedure.
LIST OF REFERENCES


Appendix A IEEE

A figure from this work is located in Chapter 2.

- Kinematic Calibration By Means Of A Triaxial Accelerometer.pdf
Appendix B Characterizing Suitability of Wearable Sensors for Movement Analysis Using a Programmed Robotic Motion

An image in this work is located in Chapter 2.

On Oct 31, 2014, at 12:11 PM, "Daniel Ashley" <dashley2@mail.usf.edu> wrote:

Hello Dr. Carey,

I would like to get copy write clearance to use an image from the paper, "Characterizing Suitability of Wearable Sensors for Movement Analysis Using a Programmed Robotic Motion" in my thesis document "Using Embedded Systems to Determine the Configuration of a Static Wheelchair Mounted Robotic Arm".

All I need is a reply to this email granting permission.

Thanks,
- Daniel Ashley
Appendix C Maximizing Manipulation Capabilities of Persons with Disabilities Using a Smart 9-Degree-of-Freedom Wheelchair-mounted Robotic Arm System

Images and figures in this work are located in Chapter 4 and Chapter 5.

---

Alqasemi, Redwan

12:37 PM (2 minutes ago) 🌟

to me 📣

Daniel,

Permission granted. Please reference the figures you used in your thesis to the paper/thesis/dissertation you took it from.

Redwan Alqasemi, Ph.D.
Research Professor, Department of Mechanical Engineering
Center for Assistive, Rehabilitation & Robotics Technologies (CARRT)
University of South Florida
4202 E. Fowler Avenue, ENB 118
Tampa, FL 33620
+1(813)974-2115

From: Daniel Ashley [mailto:dashley2@mail.usf.edu]
Sent: Friday, October 31, 2014 12:07 PM
To: Alqasemi, Redwan
Subject: Appendix Citation Permission

Hello Dr. Alqasemi,

I would like to get copy write clearance to use figures from your dissertation, "Maximizing Manipulation Capabilities of Persons with Disabilities Using a Smart 9-Degree-of-Freedom Wheelchair-Mounted Robotic Arm System" in my thesis document "Using Embedded Systems to Determine the Configuration of a Static Wheelchair Mounted Robotic Arm".

All I need is a reply to this email granting permission.

Thanks,
- Daniel Ashley
Appendix D Freescale Application Note

A figure in this work is located in Chapter 6.

Products
For Media Outlet Use Only

The images provided on this site are intended for media, reporting and informational purposes only. The images are not intended for use in advertising or in connection with the promotion of products or services. Unauthorized use of the images may violate copyright, trademark or other laws. In order to download these images, you must agree to the terms and conditions of use provided below.

Terms and Conditions

1. You agree not to use the images for any commercial purpose.
2. YOU ACKNOWLEDGE THAT THESE IMAGES ARE PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND, INCLUDING NON-INFRINGEMENT OF INTELLECTUAL PROPERTY.
3. You acknowledge that no title or intellectual property rights whatsoever are transferred to you.
4. You agree to indemnify and hold harmless Freescale Semiconductor, Inc. from any claims, lawsuits or damages resulting from your use of the images.
5. You acknowledge that the images provided for download are protected under U.S. and international laws and treaties.
6. You agree that any claim relating to your use of the images provided for download or arising under this agreement shall be governed by the substantive laws of the State of Texas.

If you do not agree with these terms, do not download any images from this site. If you are not a member of the media, or if you would like to download images for purposes other than reporting in the media, do not download images from this site.

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners. ©2014 Freescale Semiconductor, Inc.