# DEVELOPMENT OF LOW-COST, MOBILE, AND SCALABLE MOTION TRACKING USING WIRELESS AHRS MODULES

by

Thomas Dale Taylor

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Mechanical Engineering

The University of Utah

August 2017

## The University of Utah Graduate School

## STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Thomas Dale Taylor**

has been approved by the following supervisory committee members:

| | | |
|---|---|---|
| **Stacy Morris Bamberg** , | Chair(s) | **8 Jun 2017**<br>Date Approved |
| **Donald S. Bloswick** , | Member | **5 May 2016**<br>Date Approved |
| **Andrew S. Merryweather** , | Member | **5 May 2016**<br>Date Approved |
| **Carlos H. Mastrangelo** , | Member | **8 May 2017**<br>Date Approved |
| **David E. Johnson** , | Member | **5 May 2016**<br>Date Approved |

by **Tim Ameel** , Chair/Dean of

the Department/College/School of **Engineering**

and by **David B. Kieda** , Dean of The Graduate School.

# ABSTRACT

Motion capture has applications in many fields. A need has arisen for motion capture systems that are low-cost, mobile, and intuitive. An attitude heading reference system (AHRS) calculates the global orientation of a rigid body by synthesizing the output from an array of sensors. A complete motion capture system utilizing gyroscopes, accelerometers, and magnetometers attached to the main body segments of a human is proposed.

This is accomplished by providind a low-cost calibration procedure for micro electro-mechanical system (MEMS) gyroscopes, accelerometers, and magnetometers in order to create a custom AHRS unit. The accuracy of reproducing global orientations using these AHRS units is analyzed for individual modules as well as redundant groups of AHRS nodes for increased accuracy. In order to make the system intuitive, a localization procedure for finding the locations of all AHRS units attached to the body is proposed.

Sensors were successfully calibrated to an accuracy sufficient for AHRS development. The accuracy of the AHRS units was verified and led to a functioning motion capture system. The localization procedure was verified with volunteer subjects and successfully finds the location of all attached AHRS units.

For Caroline

# CONTENTS

**CHAPTERS**

**APPENDICES**

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

Low-cost MEMS gyroscopes, accelerometers, and magnetometers can be used to build an attitude heading reference system (AHRS). An AHRS can determine global orientations of a body. By combining global orientations with the forward kinematics of a human body, a low-cost, mobile, full-body motion capture system can be created from an array of AHRS units.

A calibration routine is presented for these sensors so that they can be used to create a reliable AHRS node. The accuracy of synthesizing all of the available sensor data to create an AHRS is validated. Finally, an algorithm to localize an arbitrary number of AHRS units randomly placed around the body is proposed to increase the usability of the system.

## 1.1 Project motivation

Motion capture activities that can not use a dedicated motion capture studio require a new paradigm of system. A motion capature system that is completley mobile and is completely contained on the wearer's body fixes these limitations.

### 1.1.1 Overview of AHRS motion capture

An attitude heading reference system (AHRS) is a device that uses an array of sensors to provide an orientation in the global coordinate frame. A typical approach is to use a 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer to fuse the data and provide a global orientation [1], [2]. Accelerometers are sensors that detect local accelerations (including that of gravity). Gyroscopes detect rotational velocities about the local coordinate axes. Magnetometers detect local magnetic fields and are often used as a digital compass. By combining these sensor values and using gravity and Magnetic North as a global reference, the global orientation of any body the AHRS can be determined [3]–[5].

Motion capture is the recording of motion over time and many types of systems have been developed to accomplish this. These motion capture systems have uses in a wide range of fields [6] such as three-dimensional (3D) animation [7], game development [8], rehabilitation [9], sports training [10], [11], and ergonomics [12], [13]. Human motion capture using optical systems (e.g, Vicon® [14]) that use a set of cameras with active or passive markers placed on body segments are the most common forms of motion capture due to their high accuracy and resolution. These systems typically use the cameras to determine the 3D position of the markers on the body by the two-dimesnional (2D) disparity between cameras.

Although these optically-based systems provide a high degree of accuracy, they have limitations that prevent their use for specific environments. First, optical motion capture systems such as Vicon® are cost-prohibitive for a wide range of consumer-level applications, often pushing into tens-of-thousands of dollars [14]. Second, special clothing is frequently required to provide enough contrast to the cameras that an accurate 2D tracking can be performed. Third, these systems require grounded cameras that must be set up before the system is used, and must typically be indoors to be away from sunlight. This often requires a dedicated space such as a studio. Fourth, a dedicated technician is usually needed to process and evaluate the data due to the inherent complexity of these systems.

All of these limitations provide an opportunity for an entirely different type of motion capture. Consumers or companies with access to limited capital are priced out of many of these optical systems. If special clothing or equipment is already necessary for the motion to be recorded such as athletics training or specialized ergonomics research, a motion capture suit required by the system may not be feasible. If the nature of the motions being recorded require access to specific locations such as a mountain (skiing) or a golf course, the setup of optical cameras may be time or cost prohibitive or even impossible.

These specific groups and many others will benefit by having a system that is low-cost, mobile, and usable on or under existing clothing. This dissertation addresses the question of whether AHRS motion capture can provide that solution. AHRS modules are orders of magnitude less costly than optical systems, and can be used in any environment where humans are comfortable. By attaching AHRS modules to sections of the human body, the global orientation of each section can be used to recreate the pose of a user. Calculating the forward-kinematics of each body segment allows the location of the end segments such as the hands and feet to be determined, and is performed in real-time without the need for technical analysis. The output of all attached AHRS modules can be recorded by a base system such as a computer or smartphone and the motion of a person wearing the nodes can be captured.

### 1.1.2 Specific motivations for this project

The research presented in this document is motivated by the need for a low-cost, scalable, and mobile motion capture platform. Custom-built AHRS modules are an intriguing solution to this need, but their infancy in the field of motion capture requires research to make a system that is effective, but intuitive enough to be used by consumers. The appearance of low-cost MEMS accelerometers, gyroscopes, and magnetometers from companies such as microchip®, InvenSense®, and Texas Instruments® allow the development of an inexpensive AHRS module to be developed, but the efficacy of the module as it pertains to motion capture needs to be studied.

A calibration procedure for the accelerometers, gyroscopes, and magnetometers must be developed. The MEMS sensors used in the AHRS modules output a 16-bit integer. A low-cost software-based approach to sensor calibration is needed to convert these numbers into real-world units so that they can be more readily understood. Additionally, the global declination of the Earth's magnetic fields change based on location and time of year. A magnetometer calibration procedure that is easy to use and successful at removing hard- and soft-iron magnetic field distortions is required to make an AHRS motion capture system that can adapt to new environments.

The size and low-cost of these MEMS-based AHRS modules allows redundant sensors to be placed on body segments, which may result in higher accuracy. The accuracy of single and redundant AHRS nodes needs to be researched to determine the ability of an AHRS-based mobile motion capture system to measure joint angles.

Each AHRS unit is a wireless module that must be linked to the correct body segment in order to properly compute the forward kinematics of the user. If redundant nodes can be shown to increase accuracy, a way to correctly place a high number of AHRS nodes is required. Proper placement of AHRS nodes currently requires placement according to a predetermined setup or manual entry of the location of each unit. For large sets of modules these solutions are cumbersome. An automatic localization technique to identify the body segments of all attached nodes will make full-body AHRS motion capture with redundant nodes possible.

## 1.2 Research objectives

The goal of this research project was to investigate whether AHRS nodes attached to body segments are capable of providing low-cost, accurate motion capture without requiring sophisticated technical knowledge. The following objectives answer important questions or solve crucial problems in making this system:

1. Create an inexpensive software-based calibration procedure for MEMS accelerometers, gyroscopes, and magnetometers. Ideally, the procedure may be performed by a consumer; global magnetic fields are different based on location and time of year.

2. Determine the accuracy of individual AHRS nodes and discover possible accuracy gains by combining the outputs of multiple redundant nodes.

3. Build a localizing feature that allows arbitrary placement of AHRS nodes without needing to manually input node identification in the system or require the user to follow a predetermined placement plan.

Successful implementation of these objectives is a necessary first step towards providing a mobile motion capture system.

## 1.3 Key contributions

Before the AHRS nodes can be used as a rotational reference for motion capture, the sensors must be calibrated. Each sensor outputs a raw 16-bit integer instead of a value with real-world units. Accelerometers should have units of $m/s^2$ and gyroscopes should have units of $rad/s$. Magnetometers should output a unit-less vector, but it should properly aligned with the Earth's magnetic field.

A software-based calibration technique was created to convert raw integer values from each 3-axis sensor and convert them into their real-world values. The calibration technique uses a Gauss-Newton fitting algorithm to constrain the data to an equation that describes the ideal sensor output.

The accelerometers were calibrated by collecting data while the sensor was left static in multiple positions and fitting the data to a vector of magnitude $9.81 m/s^2$. The gyroscopes were calibrated by rotating the gyroscope a known angle about each local axis and translating the raw data so that integration of the gyroscopes over time leads to the same known angle. Magnetometers are perturbed by both hard- and soft-iron distortions that cause the vector output to change magnitude depending on heading. An ellipsoid solver was created to properly translate and scale the raw data to a vector with unit magnitude. All calibrated data were compared with encoder references to validate the calibration procedure. The calibration procedure and results are presented in Chapter 2.

Once confidence in the sensor outputs have been confirmed, the accuracy of a single AHRS unit needs to be established. AHRS motion capture is completely reliant on having accurate global orientations from the attached AHRS nodes. The output of the calibrated MEMS sensors were fused together to provide a reliable global orientation of an AHRS unit.

The calibrated gyroscopes were integrated to find the change in global orientation with time. Numerical integration drifts from the correct value over time unless it is periodically corrected. The accelerometers and magnetometers were used as a reference to the gyroscopes to correct for this drift.

The accelerometers were used to find direction of gravity in the AHRS module's local coordinate frame. Similarly, the magnetometers detected the direction of Magnetic North in the node's local coordinate frame. Together, these two perpendicular directions formed a basis that acted as a reference for the gyroscopes. Because accelerometers detect local accelerations as well as the direction of gravity, fast movement could affect the resultant accelerometer heading. This required a controller to be employed that would still counteract the drift of the gyroscope integration while still preserving the integrity of the accelerometer-magnetometer pair.

Given a successful calibration procedure, the accuracy of single and redundant AHRS nodes was investigated. Individual AHRS nodes were attached to a multiple link mechanical leg with attached encoders. The global orientation output of the attached nodes were compared with the output from the mechanical leg's encoders to compare the accuracy of each node.

Once the accuracy of one AHRS node was established, more AHRS nodes were added to the mechanical leg to determine if adding redundant AHRS units could provide greater accuracy for purposes of motion capture. A method to average redundant nodes was created to improve the overall accuracy of the global orientation.

Findings from this research are presented in Chapter 3 as a conference paper. This paper was later extended to a journal article which is found in Chapter 4.

Finally, the location of AHRS modules must be established in order to set the global orientation to the correct body segment. Rather than require users of the motion capture system to correctly place or manually input the location of every attached AHRS unit, a localization procedure was devised.

The localization procedure, or *Localizer*, used reference poses that were mimicked by a user wearing AHRS nodes placed at arbitrary locations. By comparing the output of the AHRS modules with the values of the reference poses, *Probability Maps* were created that estimate the probability that any attached node is located at any possible body segment. These Probability Maps were normalized across rows so that one Probability Map could be multiplied to another and cause the locations of the AHRS nodes to converge to a location.

A small group of subjects mirrored a bank of reference poses to determine the viability of the Localizer. After it was shown that the Localizer could determine the correct locations of arbitrarily placed AHRS nodes, an analysis was performed to determine the minimum number of poses necessary to arrive to the correct solution of all nodes across all subjects. Chapter 5 explains the Localizer algorithm as well as the results of finding the minimum number of poses that are required to localize a group of AHRS nodes.

By calibrating MEMS gyroscopes, accelerometers, and magnetometers, the output of these sensors was fused together to create an inexpensive AHRS unit. The efficacy of single and redundant AHRS nodes for determining the global orientation of a rigid body was analyzed. After confidence in the AHRS modules for motion capture purposes were established, a localization algorithm was created to allow users of the system to quickly and easily use the system.

## 1.4 References

[1] T. Sakaguchi, T. Kanamori, H. Katayose, K. Sato, and S. Inokuchi, "Human motion capture by integrating gyroscopes and accelerometers," in *Mult. Fus. and Int. for Intel. Sys., 1996. IEEE/SICE/RSJ Int. Conf.* IEEE, 1996, pp. 470–475.

[2] P. Picerno, A. Cereatti, and A. Cappozzo, "Joint kinematics estimate using wearable inertial and magnetic sensing modules," *Gait & Post.*, vol. 28, no. 4, pp. 588–595, 2008.

[3] A. M. Sabatini, "Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing," *Biomed. Eng., IEEE Trans.*, vol. 53, no. 7, pp. 1346–1356, 2006.

[4] D. Roetenberg, H. Luinge, and P. Slycke, "Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors," *Xsens Mot. Tech. BV, Tech. Rep*, 2009.

[5] A. Olivares, G. Olivares, F. Mula, J. Górriz, and J. Ramírez, "Wagyromag: Wireless sensor network for monitoring and processing human body movement in healthcare applications," *J. Sys. Arch.*, vol. 57, no. 10, pp. 905–915, 2011.

[6] A. Zijlstra, M. Mancini, L. Chiari, W. Zijlstra *et al.*, "Biofeedback for training balance and mobility tasks in older populations: A systematic review," *J. Neur. Reh.*, vol. 7, p. 58, 2010.

[7] K. Pullen and C. Bregler, "Motion capture assisted animation: Texturing and synthesis," in *ACM Trans. Graph. (TOG)*, vol. 21, no. 3. ACM, 2002, pp. 501–508.

[8] E. Bethke, *Game development and production*. Wordware Publishing, Inc., 2003.

[9] H. Zhou and H. Hu, "Human motion tracking for rehabilitationa survey," *Biomed. Sig. Proc. Con.*, vol. 3, no. 1, pp. 1–18, 2008.

[10] J. R. Williams, "Athletic evaluation and training apparatus," Feb. 24 1987, U.S. Patent 4,645,458.

[11] C. H. Kim, "Method of training physical skills using a digital motion analyzer and an accelerometer," Dec. 2 1997, U.S. Patent 5,694,340.

[12] V. G. Duffy *et al.*, "A methodology for assessing industrial workstations using optical motion capture integrated with digital human models," *Occ. Erg.*, vol. 7, no. 1, pp. 11–25, 2007.

[13] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Comp. Vis. Im. Und.*, vol. 81, no. 3, pp. 231–268, 2001.

[14] (2013, October). [Online]. Available: http://www.vicon.com

# CHAPTER 2

# LOW-COST CALIBRATION OF MEMS GYROSCOPES, ACCELEROMETERS, AND MAGNETOMETERS

An attitude heading reference system (AHRS) can be created by synthesizing the output of MEMS gyroscopes, accelerometers, and magnetometers. Before this can be done, it is important to guarantee that each sensor is producing accurate readings. Sensors like these often produce a raw integer value and these values need to be converted to real-life units before they can be used. Here, a low-cost calibration routine for each sensor is proposed and the accuracy of the routine is validated.

## 2.1 Introduction

Low-cost MEMS gyroscopes, accelerometers, and magnetometers are being used in a growing number of devices. Packages such as those from Texas Instruments® [1] and Shimmer® [2] provide all-in-one access to these sensors at an inexpensive price.

A gyroscope measures rotational velocity around a defined axis. An accelerometer measures the linear acceleration along an axis and can be used to detect the constant pull of gravity or the accelerations that accompany a change in velocity. Together, gyroscopes and accelerometers are frequently packaged together as an inertial measurement unit (IMU). IMU's are frequently used in on-body tracking applications [3]–[5] and navigation/location systems [6].

Magnetometers detect the strength and direction of the local magnetic field surrounding the sensor. They can detect magnetic fields from magnetic objects or electro-magnetic devices, but are frequently used as a digital compass to detect magnetic North. Magnetometers are often used in many navigational systems [7] as attitude and heading reference systems (AHRS) [8]. They have also been used in mobile motion capture systems [9].

The output of each of these sensors typically need to be translated into real-world units to be of value. Using SI units, gyroscopes should have an output of radians per second (rad/s) or degrees per second (°/s), an accelerometer should have an output of meters per second squared (m/s$^2$), and magnetometers are frequently used as a dimensionless number that is scaled to match some reference such as the Earth's magnetic field.

The low-cost nature of these sensors have created a demand for inexpensive calibration without access to specialized tools or equipment [10]. This paper presents a low-cost calibration procedure for each of these sensors. The results of these calibration techniques were compared with reference measurements to show the efficacy of each procedure. The format of this paper is as follows. Section 2.2 provides background information on the Gauss-Newton solver algorithm, soft- and hard-iron effects on magnetometers, and the quaternion convention used in this paper. In section 2.3, methods are described in turn for each of the three sensors. The results of the efficacy of the calibration procedure are presented in section 2.4, followed by a discussion of the results in section 2.5.

## 2.2 Background

### 2.2.1 Gauss-Newton algorithm

The Gauss-Newton algorithm is a nonlinear least-squares solver and is particularly useful for fitting large data sets to nonlinear equations. Given an equation $r$ with $n$ fitting parameters $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_n)$, the Gauss-Newton algorithm iteratively solves for the minimum sum of the squares of $\boldsymbol{\beta}$. The number of data points must be greater than $n$, and $r$ should be manipulated so that it is equal to 0. For example:

$$\begin{gathered} r = 0 = \beta_0 x^2 + \beta_1 y^3 + \beta_2 z - 1 \\ \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} \end{gathered} \tag{2.1}$$

A vector $\mathbf{r}$ is created by inputing each data point in the set:

$$\mathbf{r} = \begin{bmatrix} \beta_0 {x_0}^2 + \beta_1 {y_0}^3 + \beta_2 z_0 - 1 \\ \beta_0 {x_1}^2 + \beta_1 {y_1}^3 + \beta_2 z_1 - 1 \\ \cdots \\ \beta_0 {x_m}^2 + \beta_1 {y_m}^3 + \beta_2 z_m - 1 \end{bmatrix} \tag{2.2}$$

where $m$ is the number of capture points. A Jacobian matrix $\mathbf{J}$ is constructed as follows:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial r_0}{\beta_0} & \frac{\partial r_0}{\beta_1} & \cdots & \frac{\partial r_0}{\beta_n} \\ \frac{\partial r_1}{\beta_0} & \frac{\partial r_1}{\beta_1} & \cdots & \frac{\partial r_1}{\beta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial r_m}{\beta_0} & \frac{\partial r_m}{\beta_1} & \cdots & \frac{\partial r_m}{\beta_n} \end{bmatrix} \tag{2.3}$$

For the example equation, the Jacobian is:

$$\mathbf{J} = \begin{bmatrix} {x_0}^2 & {y_0}^3 & z_0 \\ {x_1}^2 & {y_1}^3 & z_1 \\ \vdots & \vdots & \vdots \\ {x_m}^2 & {y_m}^3 & z_m \end{bmatrix} \tag{2.4}$$

Each parameter iteration is found by solving for $\mathbf{p}$:

$$\mathbf{J}^T\mathbf{J}\mathbf{p} = -\mathbf{J}\mathbf{r} \tag{2.5}$$

where

$$\mathbf{p} = \boldsymbol{\beta}^{(s+1)} - \boldsymbol{\beta}^{(s)}$$

updates the parameters $\boldsymbol{\beta}$ every iteration. The Gauss-Newton algorithm is run iteratively to solve for the parameters $\boldsymbol{\beta}$.

### 2.2.2 Soft- and hard-iron effects on magnetometers

Magnetometers measure the strength and direction of the local magnetic field. The measured magnetic field can be distorted by the environment around the magnetometer and give an incorrect heading. There are two main types of distortions to account for, hard-iron and soft-iron distortions [3].

When no distortions are present, the output of the three-axis magnetometers are a vector pointing along the magnetic field. Undistorted output will have the same magnitude and global direction regardless of orientation of the magnetometer. If the magnetometer is rotated through every possible orientation and the magnetometer data are plotted, it forms a perfect sphere centered about the origin. As an example, if a magnetometer is isolated from distortions and the resultant heading from the Earth's magnetic field is scaled to a unit vector, the magnetometer will read $(1, 0, 0)$ when the $X$-axis is pointed toward Magnetic North and will read $(0, 1, 0)$ when the $Y$-axis is pointed toward Magnetic North.

Hard-iron effects are produced by materials in the environment that exhibit a constant and additive field to the Earth's magnetic field [11]. These effects typically come from magnetic materials hard-mounted near the magnetometer. The constant magnetic field distortion shifts the output of the magnetometer away from the origin. The distorted output of the magnetometer will have the same *scaling* as the undistorted output (the output remains a sphere), but the data will be centered about a new position in space. Hard-iron effects are relatively easy to compensate for; the vector from the origin to the center of the distorted output can be subtracted from the magnetometer's output to remove these distortions.

Soft-iron effects come from environmental factors that distort the Earth's magnetic field, but do not necessarily generate their own magnetic field [11]. These effects change the observed magnetic field depending on *orientation*. Iron elements near the magnetometer or on-board wires and traces can cause these effects. These distortions are considerably more difficult to compensate for because they affect the data by scaling the data along axes that are not necessarily aligned with the magnetometer's own axes. The resultant data changes from a sphere to an ellipsoid with the principal axes rotated by a fixed value from the magnetometer's axes.

To calibrate magnetometers under soft-iron effects, the data need to be scaled *along* the principal axes of the resultant ellipsoid. This requires the distorted data to be rotated in alignment with the magnetometer's axes, scaled, and then rotated back to its original orientation. Quaternions are used to specify rotation in three-dimensional space.

### 2.2.3 Quaternion convention

There is no single convention for portraying quaternions and equations involving quaternions. In this paper, quaternions will be written as described below.

A quaternion is represented by a bold-faced **q** with the preceding subscript and superscript referring to the initial and final coordinate frame after the rotation is applied. For example, ${}^{B}_{A}\mathbf{q}$ represents a rotation from coordinate frame $A$ to coordinate frame $B$. The four parameters describing the quaternion are given as a vector.

$$\mathbf{q} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

where $x$, $y$, and $z$ represent the rotation axis of the quaternion and $w$ represents the angle of rotation.

An inverse quaternion is represented by a superscript of $-1$. This inverse is equal to a quaternion with the two coordinate frames reversed.

$$ {}^{B}_{A}\mathbf{q}^{-1} = {}^{A}_{B}\mathbf{q} $$

The $\otimes$ symbol denotes a quaternion product. Multiplying quaternions has the effect of compounding rotations. Special care to the order of the multiplication between quaternions should be taken; quaternion multiplication is not commutative. In other words, $\mathbf{q}_1 \otimes \mathbf{q}_2 \neq \mathbf{q}_2 \otimes \mathbf{q}_1$. Multiplications can be chained to move from one major coordinate frame another via a series of sub-coordinate frames:

$$ {}^{D}_{A}\mathbf{q} = {}^{B}_{A}\mathbf{q} \otimes {}^{C}_{B}\mathbf{q} \otimes {}^{D}_{C}\mathbf{q} $$

## 2.3 Methods

Calibration of the gyroscopes, accelerometers, and magnetometers are done on an individual basis. It is assumed that the sensor outputs a raw value for each global axis and that the gyroscope is captured with a timestamp. The ideal output of each sensor is as follows:

**Gyroscopes** The correct angular velocity in rad/s when in motion; a value of 0.0 rad/s when stationary.
**Accelerometers** A vector with magnitude of 9.81 m/s$^2$ when the device is stationary.
**Magnetometers** A vector with magnitude of 1.0 when the device is in any orientation.

### 2.3.1 Gyroscopes

#### 2.3.1.1 Calibration procedure

A rigorous gyroscope calibration involves spinning the unit about each axis at differing known velocities. However, when there is no access to the necessary equipment, an angular position-based integration technique can be used to get an adequate calibration. This involves attaching the gyroscopes to a rigid cube and rotating the cube around each axis by known angular amounts. The following gyroscope calibration method is intended for a single rotation around each major axis. Both the positive and negative direction is required for each axis for a total of six directions. If a more robust calibration is necessary, the method can be easily extended to solve for multiple known angles rotated at different speeds.

The gyroscope data must be collected with a timestamp for each data point. Euler's integration method is used to calculate angular position given angular velocities:

$$\theta_{x,t_1} = \theta_{x,t_0} + \omega_{x,t_0}\Delta t \tag{2.6}$$

Linear parameters are added to modify the raw gyroscope data to have units of rad/s. Here, $gyro_{x,t}$ is the raw output of the gyroscope in the $X$-direction at time $t$.

$$\omega_{x,t} = \frac{gyro_{x,t} - m_x}{\delta_x} \tag{2.7}$$

where $m_x$ and $\delta_x$ are the calibration parameters for the $X$ axis. The total angular change from $\theta_0 = 0$ to $\theta_f$ is found by summing the small angular changes from 2.6.

$$\theta_f = \frac{1}{\delta_x}\sum_{i=0}^{n} gyro_{x,t_i}\Delta t_i - \frac{m_x}{\delta x}\sum_{i=0}^{n}\Delta t_i \tag{2.8}$$

The parameters can be solved for by combining the results from two equivalent fixed rotations, one each in both the positive and negative directions:

$$\begin{aligned} m_x &= \frac{C_{1,pos} + C_{1,neg}}{C_{2,pos} + C_{2,neg}} \\ \delta_x &= \frac{C_{1,pos}}{\theta_{f,actual}} - \frac{C_{2,pos}(C_{1,pos} + C_{1,neg})}{\theta_{f,actual}(C_{2,pos} + C_{2,neg})} \end{aligned} \tag{2.9}$$

where:

$$C_{1,pos} \triangleq \sum_{i=0}^{n} x_i \Delta t_i \qquad \text{(positive rotation data)}$$

$$C_{1,neg} \triangleq \sum_{i=0}^{n} x_i \Delta t_i \qquad \text{(negative rotation data)}$$

$$C_{2,pos} \triangleq \sum_{i=0}^{n} \Delta t_i \qquad \text{(positive rotation data)}$$

$$C_{2,neg} \triangleq \sum_{i=0}^{n} \Delta t_i \qquad \text{(negative rotation data)}$$

$$\theta_{f,actual} \qquad \text{(total actual rotation of gyroscopes)}$$

Similar equations are used for the $y$ and $z$ parameters.

#### 2.3.1.2 Efficacy testing

Four calibrated 3-axis gyroscopes were attached to a mechanical 2-DOF arm that had encoders on the joints to measure the orientation of the end effector. Four 30-s trials of random end effector movement were recorded to compare the outputs of the gyroscopes to the reference orientation from the encoders. Because the gyroscopes produce an angular velocity and the encoders produce an angular position, the gyroscope output was integrated with time to produce angular position.

Numerical integration of the gyroscopes results in drift over time due to the integration of the small amount of noise naturally present in such sensors. In practice, this is addressed by techniques such as zero-velocity detection [12] which allows the integration to be stopped and restarted with motion. This practice resets the accumulation of integration errors. Here, four different intervals (1, 3, 5, and 10-s) were used for integration. After the start of each interval, the angular position was reset to the position provided by the encoder. This provided an effective periodic "rezeroing" without requiring stationary periods during the data collection, and also provided a measurement of the errors associated with long integration periods. A periodic "rezeroing" of the data was applied to the integration by setting the value of the numerical integration to the value of the encoder position at set intervals. For each trial and every direction, the numerical integration was rezeroed at 1, 3, 5, and 10-s intervals. The three axes for a single trial including all four sensors and the leg encoder are shown in **Figure 2.1**. The error between the actual orientation and the orientation calculated from the gyroscopes was calculated immediately before each rezeroing process. For example, a 30-s trial split into three 10-s intervals produces 9 measurements for comparison when each global axis is examined. Four trials were conducted, each with four sets of calibrated 3-axis gyroscopes attached. Errors from each rezeroing interval were averaged across all trials, and all global directions.

### 2.3.2 Accelerometers

#### 2.3.2.1 Calibration procedure

The 3-axis accelerometers are mounted flush to the side of a rigid cube. The raw data are collected from the accelerometers from a minimum of 6 positions ($\pm X$, $\pm Y$, $\pm Z$). Each data capture is done while the accelerometers are stationary because movement can add extra accelerations that will be unaccounted for. Five of the axes are captured by resting the bare sections of the rigid cube on a surface. The table should first be leveled using a bubble level along both directions of the surface. The data for the final axis is found by holding the face of the cube opposite the attached accelerometers on the underside of the level surface. If higher accuracy is needed, additional off-axis positions can be used as long as the accelerometers remain stationary during capture. A record of the physical orientation of the accelerometers is unnecessary.

A Gauss-Newton fitting algorithm is used to determine the parameters to fit the data to a sphere of known radius. The fitting equation is a modified form of the equation of a sphere:

$$\begin{aligned} R^2 &= x^2 + y^2 + z^2 \\ 9.81^2 &= \left(\frac{x - m_x}{\delta_x}\right)^2 + \left(\frac{y - m_y}{\delta_y}\right)^2 + \left(\frac{z - m_z}{\delta_z}\right)^2 \end{aligned} \tag{2.10}$$

The partial derivatives for each parameter needed to build the Jacobian are:

$$\begin{aligned} \frac{\partial r_i}{\partial m_x} &= \frac{-2(x_i - m_x)}{{\delta_x}^2} & \frac{\partial r_i}{\partial \delta_x} &= \frac{-2(x_i - m_x)^2}{{\delta_x}^3} \\ \frac{\partial r_i}{\partial m_y} &= \frac{-2(y_i - m_y)}{{\delta_y}^2} & \frac{\partial r_i}{\partial \delta_y} &= \frac{-2(y_i - m_y)^2}{{\delta_y}^3} \\ \frac{\partial r_i}{\partial m_z} &= \frac{-2(z_i - m_z)}{{\delta_z}^2} & \frac{\partial r_i}{\partial \delta_z} &= \frac{-2(z_i - m_z)^2}{{\delta_z}^3} \end{aligned} \tag{2.11}$$

Initial guesses are determined by calculating the midpoint value between the maximum and minimum values along an axis and the scaling from the midpoint to either extreme:

$$\begin{aligned} m_{x,guess} &= \frac{x_{max} - x_{min}}{2} + x_{min} \\ \delta_{x,guess} &= \frac{m_{x,guess} - x_{min}}{9.81} \end{aligned} \tag{2.12}$$

Similar equations are used for the $y$ and $z$ parameters.

#### 2.3.2.2 Efficacy testing

Four calibrated 3-axis accelerometers were attached to a mechanical 2-DOF arm that had encoders on the joints to measure the orientation of the end effector. For a 30-s trial, the end effector was slowly moved to different orientations. When the accelerometers are stationary, the calibrated output should be 9.81 $\mathrm{m/s^2}$ in the positive vertical direction (accelerometers read in the constant pull of gravity in the opposite direction). The calibrated output of the accelerometers was compared with the orientation of the end effector. Essentially, the accelerometers were tested for their efficacy as a tilt-sensor.

Because the encoders do not sense accelerations, the resultant accelerometer data was smoothed using a gaussian filter (**Figure 2.2**). Correlations between the smoothed accelerometer data and the directional heading of the end effector encoders were calculated using linear correlation ($r^2$), Pearson product-moment correlation coefficient, and Spearman's rank correlation coefficient. Spearman's correlation requires the data to be broken into monotonic functions (functions with no inflection points, i.e, the direction of the tangent to the curve must not change directions). A peak- and valley-finding function was used to break the data into monotonic intervals and the three correlations were calculated for each interval. These correlations were then time-weighted averaged for every global direction.

## 2.3.3 Magnetometers

### 2.3.3.1 Calibration procedure

Magnetometers under the influence of hard-iron and soft-iron effects produce an off-center and off-axis ellipsoid. To produce a sphere centered about the origin with radius 1.0, the data is fit to the generalized hyperbolic equation:

$$Ax^2 + By^2 + Cz^2 + Dxy + Exz + Fyz + Gx + Hy + Kz = 1 \tag{2.13}$$

Again, a Gauss-Newton fitting algorithm is used. The partial derivatives to build the Jacobian matrix are as follows:

$$\begin{aligned} \frac{\partial r_i}{\partial A} &= {x_i}^2 & \frac{\partial r_i}{\partial B} &= {y_i}^2 & \frac{\partial r_i}{\partial C} &= {z_i}^2 \\ \frac{\partial r_i}{\partial D} &= x_i y_i & \frac{\partial r_i}{\partial E} &= x_i z_i & \frac{\partial r_i}{\partial F} &= y_i z_i \\ \frac{\partial r_i}{\partial G} &= x_i & \frac{\partial r_i}{\partial H} &= y_i & \frac{\partial r_i}{\partial K} &= z_i \end{aligned} \tag{2.14}$$

Equation 2.13 can be rewritten in matrix form:

$$\mathbf{x}\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x} = 1 \tag{2.15}$$

where:

$$\mathbf{A} = \begin{bmatrix} A & D/2 & E/2 \\ D/2 & B & F/2 \\ E/2 & F/2 & C \end{bmatrix} \tag{2.16}$$

$$\mathbf{b} = \begin{bmatrix} G \\ H \\ K \end{bmatrix} \tag{2.17}$$

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.18}$$

The principle axes can be found from the eigenvectors and eigenvalues of **A**, *but only if the ellipsoid is centered about the origin*. This means that the parameters for equation 2.13 must be found twice. First to center the ellipsoid about the origin and then a second time to find the principle axes. The coordinates of the ellipsoid center can be found as follows [13]:

$$\mathbf{x}_{center} = -\frac{1}{2}\mathbf{A}^{-1}\mathbf{b} \tag{2.19}$$

Once this is done the magnetometer data are shifted by this amount (**Figure 2.3**). The parameters of equation 2.13 are recalculated with the shifted data. The eigenvectors of **A** are the directions of the principle axes of the ellipsoid [13]. The eigenvalues of **A** are used to find the lengths of the principle axes [13]:

$$length_i = \frac{1}{\sqrt{\lambda_i}} \tag{2.20}$$

Each eigenvector is normalized and multiplied by its corresponding length to create a vector from the center of the ellipsoid to a local maximum. The ellipsoid needs to be rotated to align the principle axes with the global coordinate frame axes. In order to calculate the correct rotation, the first eigenvector is assumed to be the axis that needs to be aligned with the global $X$ axis. A quaternion representing this initial rotation is calculated as follows:

$$\begin{aligned} \mathbf{p} &= \mathbf{X}_{global} \times \mathbf{eig}_0 \\ {}_0^1\mathbf{q} &= \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1.0 + \mathbf{X}_{global} \cdot \mathbf{eig}_0 \end{bmatrix} \end{aligned} \tag{2.21}$$

By definition, a quaternion needs to be normalized to be valid. Therefore, ${}_0^1\mathbf{q}$ should be normalized following equation 2.21. Each eigenvector is rotated by ${}_0^1\mathbf{q}^{-1}$ which transforms the data as seen in **Figure 2.4**. A view along the global $X$ axis shown in **Figure 2.5** reveals the second rotation that must be performed to align all of the principle axes with the global axes. This angle is found by comparing four angles found between:

1. Principle axis 2 and global axis $Y$,
2. Principle axis 2 and global axis $Z$,
3. Principle axis 3 and global axis $Y$, and
4. Principle axis 3 and global axis $Z$.

At least two angles in this list will always be equal as long as all of the angles are found in the same rotational direction (clockwise or counter clockwise). The two matching angles in the list are the desired rotation.

The order of the eigenvectors does not always produce a right-hand coordinate system. When a left hand coordinate system is produced, no rotation angle can align the principle axes of the ellipsoid with the global axes. This is easily corrected by reversing the direction of one of the principle axes. A simple test to determine whether the order of the principle axes produces a right or left hand coordinate system is to take the cross product of the first two principle axes. If the sign of this result matches the sign of the third principle axes, the order of the principle axes form a right-hand coordinate system.

This second rotation is an axis angle rotation about the global $X$ axis and is represented as a quaternion by:

$$ {}^{2}_{1}\mathbf{q} = \begin{bmatrix} \sin(\frac{angle}{2}) \\ 0 \\ 0 \\ \cos(\frac{angle}{2}) \end{bmatrix} \tag{2.22}$$

The total rotation from the raw data to the alignment of the ellipsoid principle axes and the global axes is found by multiplying the two quaternions:

$$ {}^{2}_{0}\mathbf{q} = {}^{1}_{0}\mathbf{q} \otimes {}^{2}_{1}\mathbf{q} \tag{2.23}$$

Scaling the ellipsoid to be a sphere of radius 1.0 is done in three steps. First, the centered data are rotated by **q** to align the principle axes with the global axes. Second, the data are scaled by the lengths of each principle axes (see equation 2.20). Third, the data are rotated back to its original orientation so that the quaternion used for scaling does not affect the final heading.

$$
\begin{aligned}
{}^{0}\mathbf{x}_{centered} &= \begin{bmatrix} x_{centered} \\ y_{centered} \\ z_{centered} \end{bmatrix} \\
{}^{2}\mathbf{x} &= {}^{2}_{0}\mathbf{q} \otimes {}^{0}\mathbf{x}_{centered} \otimes {}^{2}_{0}\mathbf{q}^{-1} \qquad (2.24) \\
{}^{2}\mathbf{x}_{scaled} &= \begin{bmatrix} \frac{x}{length_0} \\ \frac{y}{length_1} \\ \frac{z}{length_2} \end{bmatrix} \qquad (2.25) \\
{}^{0}\mathbf{x}_{scaled} &= {}^{2}_{0}\mathbf{q}^{-1} \otimes {}^{2}\mathbf{x}_{scaled} \otimes {}^{2}_{0}\mathbf{q} \qquad (2.26)
\end{aligned}
$$

The final output of equation 2.26 is a sphere of radius 1.0 with the magnetometer heading direction unchanged.

#### 2.3.3.2 Efficacy testing

Four calibrated 3-axis magnetometers were attached to a mechanical 2-DOF arm that had encoders on the joints to measure the orientation of the end effector (**Figure 2.6**). Four 30-s trials were conducted where the end effector was randomly moved. The output of the magnetometer was compared to the orientation of the end effector using linear correlation ($r^2$), Pearson product-moment correlation coefficient, and Spearman's rank correlation coefficient. Because Spearman's coefficient calculates the correlations of data to monotonic functions, each trial was broken into monotonic sections using a peak and valley finding algorithm. All correlation values were time-weighted averaged for all monotonic sections, for all trials, and for all global axes.

## 2.4 Results

### 2.4.1 Gyroscopes

The resultant error from gyroscope integration are given in **Table 2.1**. To show the effect of integration drift, the rotational axis of each sensor was set to the reference value at regular intervals. **Table 2.1** gives the average error of the integrated rotational position immediately before the reference value is applied. The mean and standard deviation of the rotational position error for time intervals of 1, 3, 5, and 10-s are provided. Each value is given in rad.

### 2.4.2 Accelerometers

The accelerometer were compared to the reference encoders using a linear correlation ($r^2$), the Pearson product-moment correlation coefficient, and Spearman's rank correlation coefficient. These results are found in **Table 2.2**. A time-weighted average of each monotonic section of a single 30-s trial was calculated for each sensor. The 30-s trial was conducted with slow movements to limit the number of accelerations that the reference encoders can not detect.

### 2.4.3 Magnetometers

Correlation between the magnetometers and the reference encoders were calculated and provided in **Table 2.3**. The correlation factors used are a linear correlation ($r^2$), the Pearson product-moment correlation coefficient, and Spearman's rank correlation coefficient. Each correlation value is the result of a time-weighted average of each monotonic section of each trial.

## 2.5 Discussion

Low-cost gyroscopes, accelerometers, and magnetometers are now easily available both in terms of cost and relative ease of gathering the data from ubiquitous microcontroller options. However, using these sensors requires calibration and traditional methods using high-tech equipment are not always readily available. This paper has presented the results of an easy and low-cost calibration method that is appropriate for these sensors.

### 2.5.1 Gyroscopes

**Table 2.1** shows that gyroscope integration is adequate for orientation estimation, particularly when short integration times are used. Even for long integration times (10-s) integration provides a good estimate of the relative orientation of the attached body. While more traditional calibration techniques can lead to more accurate results, the cost of the required machinery is prohibitive. The calibration method proposed in this paper opens relative orientation calculation to low-cost projects.

In robotics, the calculation of the position and orientation of the end-effector of a robotic arm is a common task. This is done by calculating the forward kinematics of all of the joints in a chain by multiplying the relative orientations of each joint down the chain. By attaching low-cost gyroscopes to the joints, these angles can now be found and calculated.

### 2.5.2 Accelerometers

Accelerometers are notoriously noisy instruments and are therefore often used as tilt sensors instead of a means of getting relative position by double integration of their output. The proposed calibration procedure was tested by comparing the tilt of the accelerometer to a reference global orientation. **Table 2.2** shows the correlations between the calculated accelerometer tilt-orientations and the reference global orientation.

A low-cost tilt sensor is valuable for balancing applications in robotics. If a robot needs to stay upright during operation despite the design requiring the robot to be top-heavy (the robotic equivalent of the inverted pendulum problem), low-cost accelerometers can be used as a reference to provide the necessary output to balance the robot.

### 2.5.3 Magnetometers

Magnetometers are particularly difficult to calibrate due to both hard- and soft-iron effects distorting the sensor's local magnetic field. These distortions, if unaccounted for, make the magnetometer a poor reference in oft-needed robotic localization algorithms.

As can be seen in **Table 2.3**, the software-based calibration technique presented in this paper gives excellent orientation accuracy. Having a strong global-heading as a reference is important in many robotic localization tasks such as simultaneous localization and mapping (SLAM). Unlike other sensors used for orientation, magnetometers do not need to be integrated, therefore, the orientation will not drift if the sensor is properly calibrated.

## 2.6 Conclusion

In this paper, we propose an easy and low-cost method of calibrating inexpensive MEMS gyroscopes, accelerometers, and magnetometers. For an engineer building a robot, such sensors can be readily obtained and integrated with a microcontroller to access this data; however, in order for this data to be useful, the data need to be calibrated. Using our proposed technique, the gyroscope data can be used to determine the relative joint angle of a chain of joints. The accelerometer data can be used to determine the direction of gravity for balancing purposes. The magnetometer can be used to correctly orient the robot to a map or localize the robot.

## 2.7 References

[1] T. Instruments. (2017). [Online]. Available: http://www.ti.com

[2] S. Research. (2017). [Online]. Available: http://www.shimmer-research.com/

[3] J. Vasconcelos, G. Elkaim, C. Silvestre, P. Oliveira, and B. Cardeira, "Geometric approach to strapdown magnetometer calibration in sensor frame," *Aero. and Elec. Sys., IEEE Trans.*, vol. 47, no. 2, pp. 1293–1306, 2011.

[4] E. M. Foxlin, M. Harrington, and Y. Altshuler, "Miniature six-dof inertial system for tracking hmds," in *Aero./Def. Sens. Cont.* International Society for Optics and Photonics, 1998, pp. 214–228.

[5] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *Comp. Graph. App., IEEE*, vol. 25, no. 6, pp. 38–46, 2005.

[6] S. Sukkarieh, E. M. Nebot, and H. F. Durrant-Whyte, "A high integrity imu/gps navigation loop for autonomous land vehicle applications," *Rob. and Auto., IEEE Trans.*, vol. 15, no. 3, pp. 572–578, 1999.

[7] J.-H. Wang and Y. Gao, "A new magnetic compass calibration algorithm using neural networks," *Meas. Sci. Tech.*, vol. 17, no. 1, p. 153, 2006.

[8] S. Han and J. Wang, "A novel method to integrate imu and magnetometers in attitude and heading reference systems," *J. Nav.*, vol. 64, no. 04, pp. 727–738, 2011.

[9] T. Taylor, S. Ko, C. Mastrangelo, and S. Bamberg, "Forward kinematics using imu on-body sensor network for mobile analysis of human kinematics." in *IEEE EMBS. Conf.*, vol. 2013, 2013, pp. 1230–1233.

[10] I. Skog and P. Händel, "Calibration of a mems inertial measurement unit," in *XVII IMEKO W. Cong.*, 2006, pp. 1–6.

[11] D. Gebre-Egziabher, G. Elkaim, J. D. Powell, and B. Parkinson, "A non-linear, two-step estimation algorithm for calibrating solid-state strapdown magnetometers," in *8th Int. St. P. Conf. on Nav. Sys. (IEEE/AIAA)*, 2001.

[12] I. Skog, P. Händel, J.-O. Nilsson, and J. Rantakokko, "Zero-velocity detectionan algorithm evaluation," *Biomed. Eng., IEEE Trans.*, vol. 57, no. 11, pp. 2657–2666, 2010.

[13] B. Bertoni, "Multi-dimensional ellipsoidal fitting," *Dep. Phy., S. M. U., Tech. Rep. SMU-HEP-10-14*, 2010.

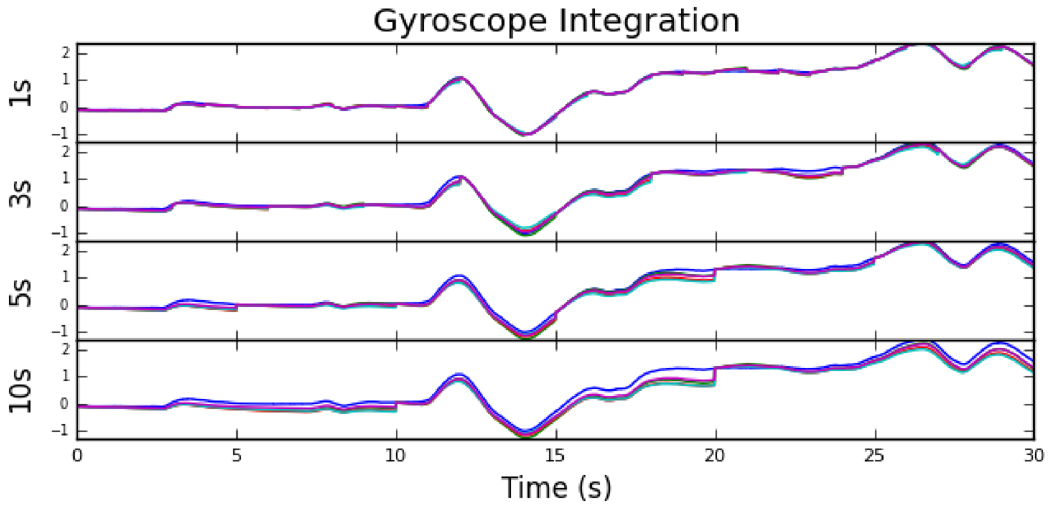

**Figure 2.1**: Error of gyroscope integration at different time intervals (1, 3, 5, and 10-s). All four sensors shown compared against encoder reference with units of radians. *X*-axis of single trial is given.

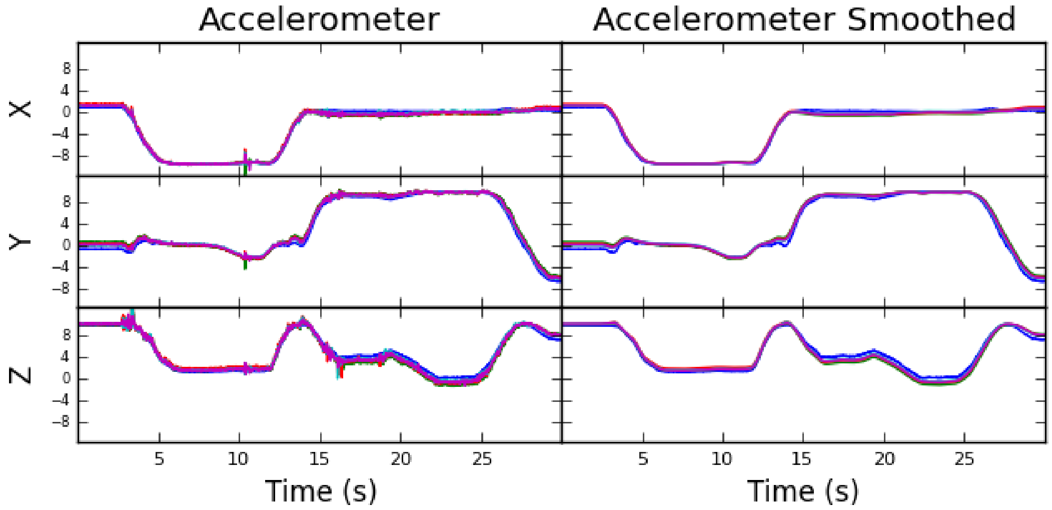

**Figure 2.2**: Comparison of calibrated accelerometer output to reference encoder data. The accelerometer data was smoothed to compare the directional tilt to the encoders attached to the mechanical leg.

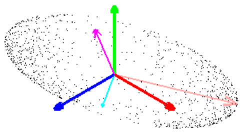

**Figure 2.3**: Magnetometer output centered about the origin. The red, green, and blue axes represent the global coordinate frame. The orange, magenta, and cyan axes are the principle axes of the ellipsoid.

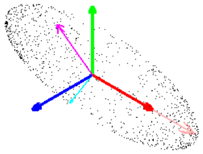

**Figure 2.4**: Magnetometer output with a rotation applied to align the first principle axis with the global $X$ axis.

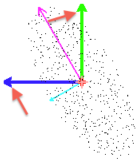

**Figure 2.5**: Side view of the first rotation. The rotation needed to align the two remaining principle axes with the global $Y$ and $Z$ is indicated by the two matching arrows.

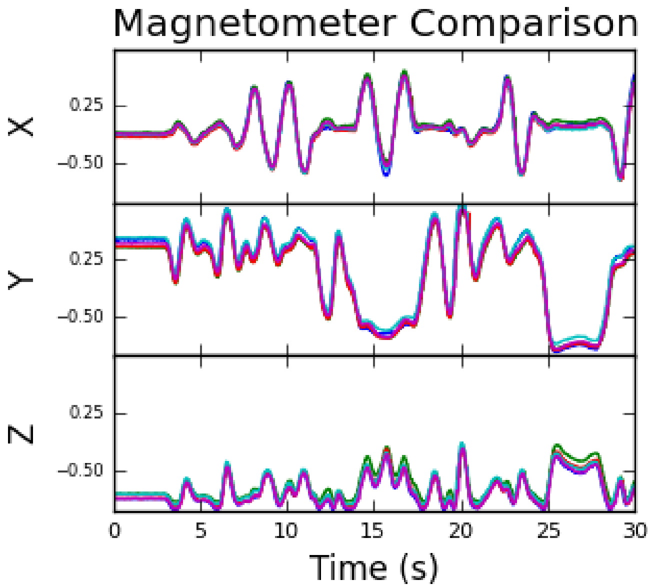

**Figure 2.6**: Comparison of calibrated magnetometer output to reference encoder data. The magnetometers are a unit-less number that only describes the heading.

**Table 2.1**: Gyroscope integration average error.

| Sensor | 1 sec (rad) ($n = 360$) | 3 sec (rad) ($n = 120$) |
|---|---|---|
| 1 | 0.016 ± 0.081 | 0.040 ± 0.112 |
| 2 | 0.014 ± 0.063 | 0.043 ± 0.114 |
| 3 | 0.016 ± 0.066 | 0.049 ± 0.116 |
| 4 | 0.011 ± 0.055 | 0.032 ± 0.097 |
| Sensor | 5 sec (rad) ($n = 72$) | 10 sec (rad) ($n = 36$) |
| 1 | 0.069 ± 0.141 | 0.137 ± 0.217 |
| 2 | 0.074 ± 0.150 | 0.147 ± 0.246 |
| 3 | 0.083 ± 0.152 | 0.167 ± 0.244 |
| 4 | 0.056 ± 0.125 | 0.111 ± 0.200 |

**Table 2.2**: Accelerometer correlations.

| Sensor | $r^2$ | Pearson | Spearman |
|---|---|---|---|
| 1 | 0.837 | 0.807 | 0.718 |
| 2 | 0.864 | 0.836 | 0.755 |
| 3 | 0.840 | 0.791 | 0.722 |
| 4 | 0.846 | 0.801 | 0.717 |

**Table 2.3**: Magnetometer correlations.

| Sensor | $r^2$ | Pearson | Spearman |
|---|---|---|---|
| 1 | 0.966 | 0.979 | 0.898 |
| 2 | 0.977 | 0.985 | 0.910 |
| 3 | 0.984 | 0.988 | 0.921 |
| 4 | 0.976 | 0.984 | 0.911 |

# CHAPTER 3

# FORWARD KINEMATICS USING IMU ON-BODY SENSOR NETWORK FOR MOBILE ANALYSIS OF HUMAN KINEMATICS

Engineering in Medicine and Biology Society Conference volume 2013, pages 1230-1233. Forward Kinematics Using IMU On-body Sensor Network for Mobile Analysis of Human Kinematics. T. Taylor, S. Ko, C. Mastrangelo, and S. Bamberg. ©Owned by the authors, published by IEEE, 2013. With permission from the Institute of Elctrical and Electronics Engineers (IEEE).

# Forward Kinematics Using IMU On-body Sensor Network for Mobile Analysis of Human Kinematics

Thomas Taylor[1], Seungoh Ko[2], Carlos Mastrangelo[3] and Stacy J. Morris Bamberg[4]

***Abstract*— The feasibility of large network inertial measurement units (IMUs) are evaluated for purposes requiring feedback. A series of wireless IMUs were attached to a human lower-limb laboratory model outfitted with joint angle encoders. The goal was to discover if large networks of wireless IMUs can give realtime joint orientation data while still maintaining an acceptable degree of accuracy.**

## I. Introduction

Nearly 10 million, or 5.2%, of U.S. adults between 18-64 years are classified with an ambulatory (walking) disability [1]. Problems in the motor and/or sensory systems can result in impaired ambulation, a disability that is generally associated with a decreased proprioceptive sense, i.e. not knowing where the limbs are or what sensory information the limbs are receiving. In order to observe motion outside the home, devices containing accelerometers, and in some cases gyroscopes and/or magnetometers have been widely used, especially as wearables for older adults [2]–[10]. A remaining challenge is evaluation in real-life settings [11]. The need for augmented feedback to improve mobility has been demonstrated through the use of Lokomat for children with cerebral palsy, an area that demonstrates the need for augmented feedback [12]. A recent review found that most studies of such devices (including commercial) had limited effectiveness, in large part due to required in-clinic use, suggesting future studies should allow feedback ”be applied during a prolonged period of time,” e.g. at home [13].

The broad appeal and low-cost nature of accelerometers has resulted in many implementations, including commercial packages such as those from Shimmer [14] and Texas Instruments [15]. Units such as these have high appeal for evaluating and providing feedback to patients in the home environment. Other applications could involve sports analysis or even actors in settings where motion analysis cameras are difficult to deploy.

This paper introduces our analysis of multiple units, which each have 3-axis accelerometers, 3-axis gyroscopes, and 3-axis magnetometers, mounted on the same rigid segments. A laboratory model of a lower-limb instrumented with encoders is used for evaluation, with three units on the tibia and three on the foot. The feasibility of large network inertial measurement units (IMUs) for motion capture purposes are evaluated by moving the lower-limb laboratory model randomly and determining if the attached IMUs can adequately capture and send data to a host computer for realtime feedback.

## II. Background

Each IMU contains a 3-axis gyroscope, a 3-axis accelerometer, and a 3-axis magnetometer. All orientation measurements are calculated from the 3-axis gyroscope which returns a 16-bit value of the angular velocity of each axis. A direction cosine matrix (DCM) is created to store the absolute orientation of the IMU based on numerical integration of each gyroscope [16]. Gyroscopes have noticeable drift over time and numerical integration can produce significant error over stretches of time; to counteract this, the accelerometer and magnetometer are used to keep the device centered. Each IMU is also equipped with an XBee$^{\mathrm{TM}}$wireless module to send periodic transmissions to a host computer for data collection.

Calculation of the IMU’s rotation matrix is created with a version of Euler’s Method of integration. The initial rotation matrix of the IMU is determined during initialization and each subsequent rotation matrix is determined from [16]:

$$\begin{aligned}
\mathbf{R}(t+dt) &= \mathbf{R}(t)\begin{bmatrix} 1 & -d\theta_z & d\theta_y \\ d\theta_z & 1 & -d\theta_x \\ -d\theta_y & d\theta_x & 1 \end{bmatrix} && (1)\\
d\theta_x &= \omega_x dt && (2)\\
d\theta_y &= \omega_y dt && (3)\\
d\theta_z &= \omega_z dt && (4)
\end{aligned}$$

where $\omega$ is the angular velocity about each axis and $dt$ is a predetermined time step. Essentially, each new rotation matrix is computed by taking the previous rotation matrix and rotating it by small angles about each axis.

To correct for error, a few techniques are used. Because each axis should remain orthogonal to each other, “renormalization” can be used to enforce this condition [16]. First the X and Y rows of the computed matrix are pulled off in to separate vectors. The dot product of these two axis should be zero because they are orthogonal. This allows a vector error to be determined:

$$error = \mathbf{X}\cdot\mathbf{Y} \qquad (5)$$

[1] Thomas Taylor is a student of Mechanical Engineering, University of Utah, 72 S. Central Campus Drive, Salt Lake City, UT 84112 `thomas.d.taylor at utah.edu`

[2] Seungoh Ko is a student of Electrical Engineering, University of Utah, 72 S. Central Campus Drive, Salt Lake City, UT 84112 `koseungoh at gmail.com`

[3] Carlos H. Mastrangelo is a USTAR professor of Electrical and Computer Engineering, University of Utah, 72 S. Central Campus Drive, Salt Lake City, UT 84112 `mastrang at eng.utah.edu`

[4] Stacy Bamber is an associate professor of Mechanical Engineering, University of Utah, 72 S. Central Campus Drive, Salt Lake City, UT 84112 `stacy.bamberg at utah.edu`

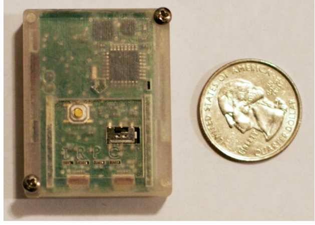

Fig. 1. IMU sensor with attached XBee$^{\text{TM}}$wireless transmitter.

This error is then appropriated between the two axis by half and the Z axis is found using the cross product:

$$
\begin{aligned}
\mathbf{X}_{orthogonal} &= \mathbf{X} - \tfrac{error}{2}\mathbf{Y} && (6)\\
\mathbf{Y}_{orthogonal} &= \mathbf{Y} - \tfrac{error}{2}\mathbf{X} && (7)\\
\mathbf{Z}_{orthogonal} &= \mathbf{X}_{orthogonal} \times \mathbf{Y}_{orthogonal} && (8)
\end{aligned}
$$

The final step is to renormalize the vectors to assure that the magnitude of each vector is one. To decrease computation time, a Taylor's expansion can be done resulting in [16]:

$$
\begin{aligned}
\mathbf{X}_{normalized} &= \tfrac{1}{2}(3 - \mathbf{X}_{orth} \cdot \mathbf{X}_{orth})\mathbf{X}_{orth} && (9)\\
\mathbf{Y}_{normalized} &= \tfrac{1}{2}(3 - \mathbf{Y}_{orth} \cdot \mathbf{Y}_{orth})\mathbf{Y}_{orth} && (10)\\
\mathbf{Z}_{normalized} &= \tfrac{1}{2}(3 - \mathbf{Z}_{orth} \cdot \mathbf{Z}_{orth})\mathbf{Z}_{orth} && (11)
\end{aligned}
$$

These steps can produce accurate results for the direction cosine matrix from the gyroscope readings. The previously mentioned drift problems that come from using numerical integration and gyroscopes prevent completely accurate results. To correct for this, the accelerometer and magnetometer are used to zero the gyroscopes. The accelerometer and magnetometer can each give an axis that does not drift that can be used as a reference. For this particular setup, the accelerometer was used for the X and Y axis and the magnetometer was used to correct for rotations about Z. To fix the drift, a simple PI controller feedback loop was used. This allowed the corrections to be tuned such that the drift cancelation was fast and accurate.

## III. Methods

To test the accuracy of both individual and redundant IMUs, a mechanical leg approximating motion of a human leg was built. This leg consists of a 3-DOF joint to simulate the joint connection to the hip, a 1-DOF knee joint, and a 2-DOF ankle joint. The mechanical leg's femur bone consists of a rigid, adjustable, telescoping aluminum rod to use the skeleton for different body sizes. Likewise, the mechanical leg's tibia bone is made from the same telescoping rod. The design of the leg allows for quick adjustment to accommodate different anthropometry values that may be of interest. Each joint has an attached magnetic encoder that returns a 10-bit analog signal. Accuracy of the magnetic encoders are approximately 0.35 degrees when operated at room temperature. For this paper, the femur bone was grounded horizontal to the floor and only the motion of the tibia and foot were analyzed. This decision was made to decrease complexity and focus on the accuracy of the IMUs instead of the forward kinematics of the leg itself.

To test the accuracy of each IMU (Fig. 1) as well as the benefit of multiple IMUs, three IMUs were attached to both the tibia and the foot. The IMUs were mounted with the IMU aligned with the bone (from knee to ankle and then from ankle to toe) while the IMU's normal vector pointed toward the ground if the leg and foot were to be extended out horizontally. Each IMU was programmed to send out information packets (consisting of roll, pitch, and yaw Euler angles as well as a time stamp synchronized across IMUs) only when the sensor saw a degree change greater than one-half. This allows the IMUs to be more energy efficient and sleep when not in use. In order to ensure efficient delivery of the data, data was transmitted as two-byte numbers for each angle and a four-byte number for the elapsed time in milliseconds. By using bytes for the angle data, resolution is lost to conserve space, but this resolution is still finer than the half a degree change needed to wake the IMU from sleep so no data is lost.

After data collection, the host computer steps through the data and linearly interpolates each sensor so that there is data from each IMU at every time step. Because of the sleep-interrupt-wake feature employed in the IMUs to preserve battery life, data does not come in at predictable intervals. By doing a sensor by sensor linear interpolation, this problem is addressed. In a separate process, another program gathered readings from the artificial leg's joint encoders, again with an attached time stamp.

The data collection process is as follows:

1) Turn on each sensor and zero all accelerometer and gyroscope readings.
2) Connect each sensor to the host computer and place each sensor in stasis.
3) Attach each sensor to the artificial leg in the orientation heretofore mentioned.
4) Start data collection for the leg's encoders.
5) Ping each sensor to synchronize clocks and start sending data.
6) Randomly move the joints of the leg for about twenty seconds.
7) Linearize the IMU data.
8) Apply calibration to magnetometer

The magnetometer works much like a compass. It has an internal spindle that aligns itself with the nearest magnetic field and it is hoped that this magnetic field is pointing toward Magnetic North. If there are small local magnetic fields which can be present from running electronic equipment in the room or even the room's structural material, the magnetometers can produce an incorrect alignment. To make matters worse, each magnetometer for each IMU does not have a factory calibration done by the manufacturer so a batch of magnetometers can behave differently.

To overcome this obstacle of misaligned magnetometer headings, the IMUs can be centered locally by knowing the

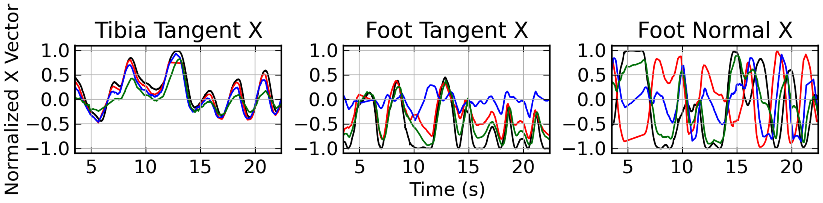

Fig. 2. Raw x component data for vectors calculated from the IMU. Note that although there are only 3 sensors per joint (Tibia and Foot), the foot has enough degrees of freedom to require both a vector that is tangent to the foot and a vector normal to the foot to be calculated. The black curve represents the true vector heading and the red, green, and blue curves represents output from the three sensors per joint.

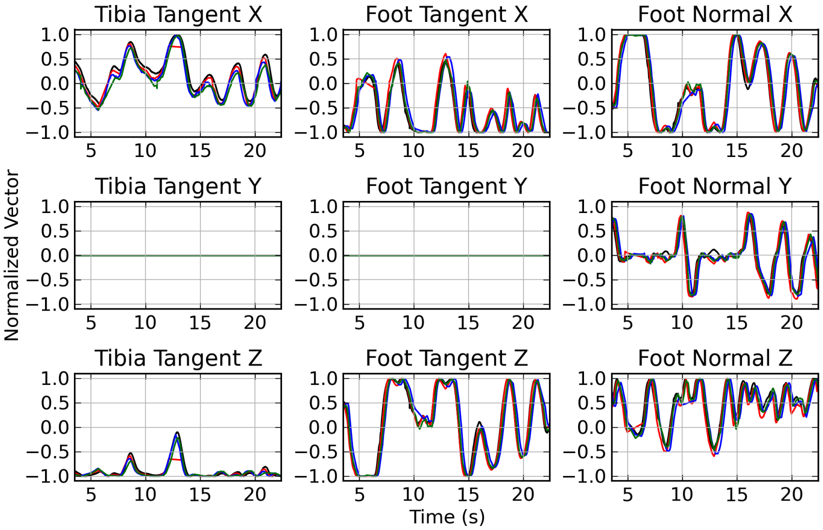

Fig. 3. Output from calibrated sensors. Three sensors are attached to each joint and full x, y, and z components of the calculated vectors are shown. The foot joint has enough degrees of freedom that two vectors (a tangent and normal) are needed. The black curves represent the true values of each joint and the red, green, and blue curves represent outputs from the sensors.

true heading of the body the IMU is attached to. Much like a scale can be calibrated by weighing a mass of known value, the IMUs can be post calibrated by knowing the true tangential vector heading. The leg was positioned so that it points toward Magnetic North which means the tangential vector of the IMU must lie in the X-Z plane. By finding the angle the magnetometer has shifted from this heading, the IMU can be rotated back to the correct heading.

The first step is to find out how far out of alignment the IMU is oriented. To do this, the X and Y components of the heading vector are considered. By finding the tangent of the two components, the angle deviation from the X-axis can be determined. Note that the graphical quadrant the resultant vector is located in is very important; a resultant vector in quadrant II for instance must be rotated *counter-clockwise* to be in alignment with the X-axis. The tangent function has a reduced range because $\tan(\frac{y}{x})$ yields the same result as $\tan(\frac{-y}{-x})$. This shortcoming is resolved by using the *ATAN2* function present in most programming language math libraries. This function expands the range of the tangent function to $[-\pi, \pi]$ with the knowledge of what the signs of $x$ and $y$ are.

After the quadrant is determined, the sensor can be re-calibrated and pulled back to the X-axis. This is done with a two-dimensional vector rotation:

$$x' = x\cos\theta - y\sin\theta \tag{12}$$

$$y' = x\sin\theta + y\cos\theta \tag{13}$$

where $\theta$ is the rotation of the vector in a counter-clockwise direction, $x$ and $y$ are the original vector components and $x'$ and $y'$ are the components of the new rotated vector. The rotation logic then becomes:

- If in Quadrant I, rotate the vector by $-\theta$
- If in Quadrant II, rotate the vector by $\pi - \theta$
- If in Quadrant III, rotate the vector by $-\pi - \theta$

- If in Quadrant IV, rotate the vector by $-\theta$

The normal vector of the foot is a little more complicated to correct because there is not a single axis to correct against. The two compounded ankle rotations allows the normal vector to move in all three axis. To correct for this vector, first the IMU is correctly oriented along the axis of the foot. Secondly, the forward vector is rotated $-\frac{\pi}{2}$ about the IMU's Y-axis. This new vector is the IMU's normal vector. Finally, this normal vector is rotated about the IMU's tangential axis by the value of *roll* - one of the three Euler Angles - which is output from the sensor.

$$up'_y = up_y \cos roll - up_z \sin roll \tag{14}$$

$$up'_z = up_y \sin roll + up_z \cos roll \tag{15}$$

## IV. Results

Results from the raw data are shown in Fig. 2. Each plot shows the x-component of either the tangent or normal vector of each IMU as well as where the true heading of this component should be as calculated from the artificial leg's encoders. Error from the IMU magnetometers are visible in each plot with particular noise found in the IMUs mounted to the foot. Fig. 3 shows the data after being calibrated using the *ATAN2* method.

## V. Discussion

Fig. 3 shows data that is significantly corrected from the raw IMU data. Instead of the noisy data seen in Fig. 2, data is consistent across IMUs and matches the expected values calculated from the artificial leg. Some errors are still present, for instance in *Tibia Tangent X* and *Tibia Tangent Z*, one of the IMUs cuts across a "hump" around the 12 second mark. This occurs because this particular sensor did not send out data for a brief period of time. Because all of the IMU data is linearized to provide an orientation point for all IMUs for all time, if there is a break in data, the true curve can not be correctly determined.

Also, the *Foot Normal Y* plot shows incorrect calculation at about 9 minutes and 13 seconds. Closer examination shows that the computed magnitude of the incorrect "humps" is what is expected, but the sign of the data has been flipped. When this happens, it is an indication that there is a range error corresponding to the $\cos$ and $\sin$ functions used for data analysis. Because these functions operate on a circle, it is possible for an angle to move out of bounds temporarily and produce error. More analysis is necessary to ensure that small errors like this are no longer existent.

The IMUs are sufficient to be used in real time networked applications as shown. For future work, the aforementioned calculation errors need to be resolved and larger networks of IMUs should be tested. If very large networks of IMUs are shown to be effaceable, very accurate real time motion capture data is possible without a dedicated lab. Effort should be made to apply averaging methods to the IMUs to try and arrive at more accurate and reliable information. Planned future work consists of refining a Kalman filter that would allow the user to trust data if it has arrived with certain criteria such as data that is closer to nearby IMUs or to trust data more if it comes from an IMU that pings the host computer frequently.

The method in this paper has shown that it is possible to get reasonably accurate information from IMUs and that it is possible to scale the number of IMUs present on a body and still achieve real time results. Greater networks of IMUs will need to be tried so that lower cost motion capture systems can be available outside of dedicated laboratories.

## References

[1] R. Research, T. C. on Disability Statistics, and Demographics, "Annual disability statistics compendium," 2010.

[2] S. Patel, K. Lorincz, R. Hughes, N. Huggins, J. Growdon, D. Standaert, M. Akay, J. Dy, M. Welsh, and P. Bonato, "Monitoring motor fluctuations in patients with parkinson's disease using wearable sensors," in *IEEE Transactions On Information Technology in Biomedicine*, vol. 13, November 2009, pp. 864–873.

[3] T.-T. Nguyen, M.-C. Cho, and T.-S. Lee, "Automatic fall detection using wearable biomedical signal measurement terminal," in *31st Annual International Conference of the IEEE EMBS*. IEEE, September 2009, pp. 5203–5206.

[4] M. Kishimoto, T. Yoshida, T. Hayasaka, D. Mori, Y. Imai, N. Matsuki, T. Ishikawa, and T. Yamaguchi, "An internet-based wearable watch-over system for elderly and disabled utilizing emg and accelerometer," in *Technol Health Care*, vol. 17, 2009, pp. 121–131.

[5] M. Hanlon and R. Anderson, "Real-time gait event detection using wearable sensors," in *Gait Posture*, vol. 30, November 2009, pp. 523–527.

[6] W. S. Yeoh, I. Pek, Y. H. Yong, X. Chen, and A. B. Waluyo, "Ambulatory monitoring of human posture and walking speed using wearable accelerometer sensors," in *Conf Proc IEEE Eng Med Biol Soc*, 2008, pp. 5184–5187.

[7] M. V. Wieringen and J. Eklund, "Real-time signal processing of accelerometer data for wearable medical patient monitoring devices," in *Conf Proc IEEE Eng Med Biol Soc*, 2008, pp. 2397–2400.

[8] M. N. Nyan, F. E. Tay, and E. Murugasu, "A wearable system for pre-impact fall detection," in *J Biomech*, vol. 41, December 2008, pp. 3475–3481.

[9] P. Bonato, D. M. Sherrill, D. G. Standaert, S. S. Salles, and M. Akay, "Data mining techniques to detect motor fluctuations in parkinson's disease," in *Conf Proc IEEE Eng Med Biol Soc*, vol. 7, 2004, pp. 4766–4769.

[10] B. R. Chen, S. Patel, T. Buckley, R. Rednic, D. J. McClure, L. Shih, D. Tarsy, M. Welsh, and P. Bonato, "A web-based system for home monitoring of patients with parkinson's disease using wearable sensors," in *IEEE Trans Biomed Eng*, vol. 58, March 2010, pp. 831–836.

[11] E. D. de Bruin, A. Hartmann, D. Uebelhart, K. Murer, and W. Zijlstra, "Wearable systems for monitoring mobility-related activities in older people: a systematic review," in *Clin Rehabil*, vol. 22, October 2008, pp. 878–895.

[12] B. Patritti, M. Sicari, L. Deming, F. Romaguera, M. Pelliccio, M. G. Benedetti, D. Nimec, and P. Bonato, "Enhancing robotic gait training via augmented feedback," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, 2010, pp. 2271–2274.

[13] A. Zijlstra, M. Mancini, L. Chiari, and W. Zijlstra, "Biofeedback for training balance and mobility tasks in older populations: a systematic review," in *J Neuroeng Rehabil*, vol. 7, 2010, p. 15.

[14] S. Research. [Online]. Available: http://www.shimmer-research.com/

[15] [Online]. Available: http://www.ti.com/tool/cc2541dk-sensor?DCMP=blestackHQS=blestack13-pr-tf1buy

[16] W. Premerlani and P. Bizard, "Direction cosine matrix imu: Theory," May 2009, http://gentlenav.googlecode.com/files/MahonyPapers.zip.

# CHAPTER 4

# ACCURACY OF CUSTOM AHRS ON-BODY SENSOR NETWORK FOR MOBILE ANALYSIS OF HUMAN KINEMATICS

After validating the MEMS sensors and establishing a low-cost calibration routine, the AHRS units can be used to track individual bone segments of the body. Each AHRS captures a global orientation and this information can be with the forward kinematics of a human body to determine the relative positions of body segments. The feasibility of AHRS units for a full-body motion capture system is examined here by looking at the accuracy of a set of AHRS nodes.

## 4.1 Introduction

Nearly 10 million, or 5.2% of U.S. adults between 18-64 years are classified with an ambulatory (walking) disability [1]. Problems in the motor and/or sensory systems can result in impaired ambulation, a disability that is generally associated with a decreased proprioceptive sense, i.e, not knowing where the limbs are or what sensory information the limbs are receiving. In order to observe motion outside the home, devices containing accelerometers, and in some cases gyroscopes and/or magnetometers have been widely used, especially as wearables for older adults [2]–[10]. A remaining challenge is evaluation in real-life settings [11]. The need for augmented feedback to improve mobility has been demonstrated through the use of Lokomat for children with cerebral palsy, an area that demonstrates the need for augmented feedback [12]. A recent review found that most studies of such devices (including commercial) had limited effectiveness, in large part due to required in-clinic use, suggesting future studies should allow feedback “be applied during a prolonged period of time,” e.g, at home [13].

The broad appeal and low-cost nature of accelerometers has resulted in many implementations, including commercial packages such as those from Shimmer [14] and Texas Instruments [15]. Units such as these have high appeal for evaluating and providing feedback to patients in the home environment. Other applications could involve sports analysis or even actors in settings where motion analysis cameras are difficult to deploy.

This paper introduces our analysis of multiple custom AHRSs mounted to the same rigid segments. Each AHRS has a 3-axis accelerometer, 3-axis gyroscope, and a 3-axis magnetometer. A laboratory model of a lower-limb instrumented with encoders is used for evaluation, with three units on the tibia and three on the foot. The feasibility of large network attitude and heading reference systems (AHRSs) for motion capture purposes are evaluated by moving the lower-limb laboratory model randomly and determining if the attached AHRSs can capture and send data to a host computer and have the data within 5% of the true orientation of the joint. This paper is a direct extension of [16] and adds an additional method of averaging redundant AHRSs as well as adds correlation analysis to determine overall accuracy.

## 4.2 Background

Each AHRS (**Figure 4.1**) contains a 3-axis gyroscope, a 3-axis accelerometer, and a 3-axis magnetometer. All orientation measurements are calculated from the 3-axis gyroscope which returns a 16-bit value of the angular velocity of each axis. A direction cosine matrix (DCM) is created to store the absolute orientation of the AHRS based on numerical integration of each gyroscope [17]. Gyroscopes have noticeable drift over time and numerical integration can produce significant error over long periods; to counteract this, the accelerometer and magnetometer are used to keep the device centered. This is accomplished by using the 3-axis accelerometer as a tilt sensor to ground the roll and pitch Euler angles and the magnetometer heading is used to ground the yaw angle. Each AHRS is also equipped with an XBee$^{TM}$wireless module to send periodic transmissions to a host computer for data collection.

Calculation of the AHRS's rotation matrix is created with a version of Euler's method of numerical integration [17]. Using this method, the initial rotation matrix of the AHRS is determined during initialization and each subsequent rotation matrix is determined from:

$$\begin{aligned}
\mathbf{R}(t+dt) &= \mathbf{R}(t)\begin{bmatrix} 1 & -d\theta_z & d\theta_y \\ d\theta_z & 1 & -d\theta_x \\ -d\theta_y & d\theta_x & 1 \end{bmatrix} && (4.1)\\
d\theta_x &= \omega_x dt && (4.2)\\
d\theta_y &= \omega_y dt && (4.3)\\
d\theta_z &= \omega_z dt && (4.4)
\end{aligned}$$

where $\omega$ is the angular velocity about each axis and $dt$ is a predetermined time step. Essentially, each new rotation matrix is computed by taking the previous rotation matrix and rotating it by small angles about each axis.

To correct for error, a few techniques are used. Because each axis should remain orthogonal to the other two, "renormalization" can be used to enforce this condition [17]. First, the X and Y rows of the computed matrix are extracted as separate vectors. The dot product of these two axes should be zero because they are orthogonal. This allows a vector error to be determined:

$$error = \mathbf{X} \cdot \mathbf{Y} \tag{4.5}$$

This error is then appropriated between the two axes by half and the Z axis is found using the cross product:

$$\begin{aligned}
\mathbf{X}_{orthogonal} &= \mathbf{X} - \tfrac{error}{2}\mathbf{Y} && (4.6)\\
\mathbf{Y}_{orthogonal} &= \mathbf{Y} - \tfrac{error}{2}\mathbf{X} && (4.7)\\
\mathbf{Z}_{orthogonal} &= \mathbf{X}_{orthogonal} \times \mathbf{Y}_{orthogonal} && (4.8)
\end{aligned}$$

The final step is to renormalize the vectors to ensure that the magnitude of each vector is one. To decrease computation time, a Taylor's expansion can be done [17], resulting in:

$$\begin{aligned}
\mathbf{X}_{normalized} &= \tfrac{1}{2}(3 - \mathbf{X}_{orth} \cdot \mathbf{X}_{orth})\mathbf{X}_{orth} && (4.9)\\
\mathbf{Y}_{normalized} &= \tfrac{1}{2}(3 - \mathbf{Y}_{orth} \cdot \mathbf{Y}_{orth})\mathbf{Y}_{orth} && (4.10)\\
\mathbf{Z}_{normalized} &= \tfrac{1}{2}(3 - \mathbf{Z}_{orth} \cdot \mathbf{Z}_{orth})\mathbf{Z}_{orth} && (4.11)
\end{aligned}$$

These steps can correct for errors that come from the direction cosine matrix. Because the direction cosine matrix is computed entirely from a numerical integration of the gyroscope readings, the result will drift from the true value over time. To compensate for drift the accelerometer and magnetometer are used. The accelerometer is used as a 2D tilt sensor to ground the direction cosine matrix around the X and Y axes and the magnetometer is used to ground the direction cosine matrix around the Z axis. This is accomplished by setting up a PI controller for each axis. The accelerometer and magnetometer are used as reference values and the PI controllers are tuned to correct for the slow gyro drift over time.

## 4.3 Methods

### 4.3.1 Data collection

To test the accuracy of both individual and redundant AHRSs, a mechanical leg approximating motion of a human leg was built. This leg consists of a 3 degree of freedom (DOF) joint to sAHRSlate the joint connection to the hip, a 1-DOF knee joint, and a 2-DOF ankle joint. The mechanical leg's femur bone consists of a rigid, adjustable, telescoping aluminum rod to use the skeleton for different body sizes. Likewise, the mechanical leg's tibia bone is made from the same telescoping rod. The design of the leg allows for quick adjustment to accommodate different anthropometry values that may be of interest. Each joint has an attached magnetic encoder that returns a 10-bit analog signal. Accuracy of the magnetic encoders are within ±0.35° when operated at room temperature. For this paper, the femur bone was grounded horizontal to the floor and only the motion of the tibia and foot were analyzed. This decreased the complexity of the forward kinematic calculations required to analyze the accuracy of the AHRS measurements.

Each AHRS outputs roll, pitch, and yaw Euler angles that represent the *global* (world) orientation of the node. These orientations are converted to two normalized vectors - the *forward* vector representing the heading of the AHRS and the *up* vector which describes the orientation of the top side of the AHRS with a vector normal to this face. When attached to the joints on the mechanical leg, the nodes attached to the shin have their forward vectors pointed along the shin from knee to ankle and the up vectors are oriented in the direction facing forward as if the leg is in the standing position. The AHRSs attached to the foot have the forward vectors pointed along the foot from ankle to toe and their up vectors are in the upward direction from the top of the foot. To evaluate the accuracy of the orientations calculated from the AHRSs, the encoder angles from the mechanical leg are used to calculate the true direction of each joints' *forward* and *up* vector. The true vectors are compared to the calculated node vectors by breaking the vectors into global X, Y, and Z components.

To assess the accuracy of each AHRS as well as the benefit of multiple AHRSs, three nodes were attached to both the tibia and the foot. Each AHRS was programmed to send data (consisting of the Euler angles as well as a time stamp of when the data was sent) only when the sensor saw an angle change greater than 0.5°. This allowed the AHRSs to conserve energy and sleep when not in use. The data was transmitted as two-byte numbers for each angle and a four-byte number for the elapsed time in milliseconds. By using two bytes for the angle data to efficiently transmit data, resolution was decreased; however, this resolution was still finer than the 0.5° change needed to wake the AHRS from sleep. The host computer captured data over the serial port at a constant rate of 50 Hz (20 ms).

After data collection, the host computer stepped through each sensor and linearly interpolated the data so that there were data from each AHRS at every time step. Because of the sleep-interrupt-wake feature employed in the AHRSs to preserve battery life, data do not come in at predictable intervals. By doing a sensor by sensor linear interpolation, this problem is addressed. In a separate process, another program gathered readings from the artificial leg's joint encoders, again with an attached time stamp. The encoder data arrived approximately every 12 ms.

The data collection process was as follows:

1. Turn on each sensor and zero all accelerometers.

2. Connect each sensor to the host computer and place each sensor in stasis.

3. Attach each sensor to the artificial leg.

4. Start data collection for the leg's encoders.

5. Ping each sensor to synchronize clocks and start sending data.

6. Randomly move the joints of the leg for about twenty seconds.

7. Linearly interpolate the AHRS data.

8. Apply calibration to magnetometer and perform further post-processing

### 4.3.2 Postcollection calibration

The magnetometer works much like a compass. It has an internal spindle that aligns itself with the nearest magnetic field which is ideally pointing toward Magnetic North. If there are small local magnetic fields, which can be present from running electronic equipment in the room or created from the room's structural material, the magnetometers can produce an incorrect alignment. Unfortunately, the magnetometers used did not have a factory calibration so a batch of magnetometers can behave differently. A soft iron and hard iron calibration can be done on the magnetometers [18], but this method is only a generic calibration and does not account form errors that come from local magnetic fields that come from infrastructure or nearby running equipment.

To overcome this obstacle of misaligned magnetometer headings, the AHRSs can be centered locally by knowing the true heading of the body the AHRS is attached to. Much like a scale can be calibrated by weighing a mass of known value, the AHRSs can be post-calibrated by knowing the true *forward* vector heading. The leg was positioned to point toward Magnetic North when extended which means the tibia joint only moved in the X-Z plane. By finding the angle the magnetometer had shifted from this heading, the AHRS can be rotated back to the correct heading.

The first step was to find out how far out of alignment the AHRS was oriented. To do this, the X and Y components of the *forward* vector were considered. By finding the tangent of the two components, the angle deviation from the X-axis was determined. Note that the graphical quadrant the resultant vector is located in is very important; a resultant vector in quadrant II, for instance, must be rotated *counter-clockwise* to be in alignment with the X-axis. The tangent function has a reduced range because $\tan(\frac{y}{x})$ yields the same result as $\tan(\frac{-y}{-x})$. This shortcoming is resolved by using the *ATAN2* function present in most programming language math libraries. This function expands the range of the tangent function to $[-\pi, \pi]$ with the knowledge of what the signs of $x$ and $y$ are.

After the quadrant is determined, the sensor can be re-calibrated and rotated back to the X-axis. This is done with a 2D vector rotation:

$$x' = x\cos\theta - y\sin\theta \tag{4.12}$$

$$y' = x\sin\theta + y\cos\theta \tag{4.13}$$

where $\theta$ is the rotation of the vector in a counter-clockwise direction, $x$ and $y$ are the original vector components and $x'$ and $y'$ are the components of the new rotated vector. The rotation logic then becomes:

- If in Quadrant I, rotate the vector by $-\theta$
- If in Quadrant II, rotate the vector by $\pi - \theta$
- If in Quadrant III, rotate the vector by $-\pi - \theta$
- If in Quadrant IV, rotate the vector by $-\theta$

The *up* vector of the foot was a little more complicated to correct because there was not a single axis to correct against. The two compounded ankle rotations allow the *up* vector to move in all three axes. To correct for this , first the AHRS was correctly oriented along the axis of the foot. Secondly, the *forward* vector was rotated $-\frac{\pi}{2}$ about the AHRS's Y-axis. This new vector is the AHRS's *up* vector. Finally, this vector was rotated about the AHRS's local *forward* axis by the value of *roll* - one of the three Euler Angles - which is output from the sensor.

$$up'_y = up_y \cos roll - up_z \sin roll \tag{4.14}$$

$$up'_z = up_y \sin roll + up_z \cos roll \tag{4.15}$$

### 4.3.3 Individual AHRS time shifting and averaging

Even though each sensor was pinged by the host computer to synchronize the AHRS clocks with the clock on the base computer, latency from the initial ping to the individual nodes was not consistent, which caused the data of each sensor to have incorrect timestamps. A least-squares shifting approach was used to center the data around a known movement. Because all of the AHRS nodes were attached to the laboratory mechanical leg with attached encoders, each AHRS was time-shifted to match the encoder data from the laboratory model. To determine the optimal time offset to match the leg encoders, each AHRS curve was compared to the corresponding encoder data by the sum of the squares:

$$SS_{tot} = \sum_{i=1}^{n}(y_i - \bar{y})^2 \tag{4.16}$$

$$SS_{res} = \sum_{i=1}^{n}(f_i - y_i)^2 \tag{4.17}$$

$$R^2 = 1 - \frac{SS_{tot}}{SS_{res}} \tag{4.18}$$

where $y$ is the value of the AHRS data, $f$ is the value of the laboratory model encoders, and $\bar{y}$ is the mean of the AHRS data. A program function was created to shift the encoder data back and forth to maximize the value of $R^2$. Once the time offsets that maximize the value of $R^2$ were found, all of the data were aligned.

A redundant number of AHRSs per kinematic joint can give a more accurate representation of the true joint angle by averaging and further data reduction. Simply averaging the calculated data from the redundant sensor nodes is not sufficient because of the random and intermittent transmission of data from each node. All of the AHRSs went through a postcapture linear interpolation process to ensure that there is data for every sensor at every period of time. If one of the AHRSs has not transmitted data to the host computer in a significantly long period of time, this data are unreliable and will incorrectly affect the eventual averaged data. To overcome this obstacle, the data were analyzed to find periods of time where individual AHRSs had not transmitted data. Those periods containing a long absence of data from a given sensor were thrown out while the remaining redundant sensors were averaged. If a situation arose where all redundant sensors on a joint were unavailable at the same time, the interpolated data were averaged as if all the sensors were online and a warning message was sent to the user indicating unreliable data during that period. For this particular setup, AHRSs were flagged for unreliable data when a time greater than 300ms without data transmission occurred.

### 4.3.4 Determination of AHRS accuracy

Three metrics were used to determine the accuracy of the the AHRS node network: linear correlation based on the least-squares method ($R^2$), Pearson's product-moment correlation coefficient, and Spearman's rank correlation coefficient. The least-squares correlation ranges between 0 and 1 with 1 representing a perfect correlation. For the Pearson and Spearman's coefficients, the value ranges between -1 and 1. When these correlations are calculated to be a negative value, it means that the data follows a reversed correlation (such as the base curve moving in a positive direction while the inspected curve moves in a negative direction). Spearman's correlation coefficient determines how well the relationship between two variables can fit along a monotonic function and because the data produced by the AHRSs are based on random movements, the data needed to be separated into monotonic curves.

A monotonic function is a function whose first derivative does not change signs. The curves from the calculated AHRS data have multiple inflection points. To break the data into a series of monotonic sets, a peak and valley finding algorithm was implemented. Correct identifications of peaks and valleys in the overall curve are the start and end points of monotonic curves. The location of the peaks and valleys was calculated based on the curves produced from the encoders. Because slight noise in a data set can cause errors in a peak and valley finding algorithm, the encoder data was put through a one-dimensional (1D) second-order Gaussian filter for slight smoothing. **Figure 4.2** shows a visual example of these steps.

## 4.4 Results

Results from the raw data are shown in **Figure 4.3**. Each plot shows the X-component of either the *forward* or *up* vector of each AHRS as well as where the true heading of this component should be as calculated from the artificial leg's encoders. Error from the AHRS magnetometers are visible in each plot with particular noise found in the AHRSs mounted to the foot. **Figure 4.4** shows the same data after being calibrated using the *ATAN2* method. Results from all post-processing, including averaging of redundant AHRS nodes, is shown in **Figure 4.5** for every direction for every joint. Plots of each sensor are provided in Euler angles as an appendix. Caution should be taken when making comparisons with Euler angle plots because discontinuities in the curves caused by circular rollover can make errors appear larger than they are.

## 4.5 Discussion

### 4.5.1 Calibration and averaging

The improvement from the raw AHRS data is clear in **Figure 4.4**. Instead of the noisy data seen in **Figure 4.3**, data are consistent across AHRSs and close to the expected values calculated from the artificial leg. Some errors are still present; for instance, in *Tibia Forward X* and *Tibia Forward Z*, one of the AHRSs cuts across a "hump" around the 10-s mark. This occurs because this particular sensor did not send out data for a brief period of time. Because all of the AHRS data are linearly interpolated to provide an orientation point for all AHRSs at each time point, if there is a break in data, the linearly interpolated points are unlikely to represent the true motion.

**Figure 4.6** shows the difference between the output from the individual AHRS nodes and the result after being averaged. As stated previously, simply finding the mean of a joint's redundant sensors is not sufficient because of interpolation between unpredictable data transmissions. This problem is showcased clearly in the left plot of **Figure 4.6**. Because one of the AHRSs did not transmit information between approximately 9.6 and 11.0-s, the linear averaging of interpolated data would cause the local network average to move away from the true value. To correct this, if a node has not sent a data transmission for longer than 300 ms, the data in between are disregarded and the remaining nodes are used to calculate the mean. If all of the sensors on a joint experience this problem at the same time, the time range for unreliable data is reported back to the user and all of the interpolated data in this time period are averaged. For this trial, this occurred once for the foot joint for a period of 540 ms. The right plot of **Figure 4.6** shows the output of the averaging algorithm applied. In this plot, there are tiny spikes in the average curve that occur when on of the averaged sensors suddenly drops out.

### 4.5.2 Accuracy of AHRS network

The three correlation coefficients being considered were calculated for each monotonic section. To visualize the data, **Figure 4.7** shows box plots for each sensor and each direction. Correlation coefficients far enough from the mean to be considered an outlier (represented in the figure as a "+") primarily come from sections of data where an individual node has gone offline, such as the data portrayed in **Figure 4.6**.

The box plots in **Figure 4.8** show the effect of averaging the AHRS data across redundant nodes. Most significantly, the large majority of outlier points have been eliminated. The prime source for this comes from eliminating data when an individual node has not sent a packet in a significantly long period of time. As shown in **Figure 4.8**, the only remaining section with significant error according to all three correlation coefficients is *Foot Up Y*. A visualization for this error can be seen in **Figure 4.9**. It is important to recognize that the range of the normalized vector is only 0.2 and the time length of the section of error is only about 300 ms. **Table 4.1** gives a mean value across every monotonic section of each correlation coefficient. The lowest average correlation value is Spearman's rank correlation coefficient with a value of 0.852. This averaged value includes the section portrayed in **Figure 4.9** which has a strong *negative* correlation which pulls the mean value deceptively low. These averaged correlation coefficient values are all within 5% of a perfect correlation except for *Foot Up Y* and the Spearman's value for *Tibia Forward Z*.

## 4.6 Conclusion

The network of redundant AHRSs attached to a laboratory model mechanical leg successfully output data within a 5% error of the true joint orientations. Although the raw AHRS data contain orientation errors caused from incorrect magnetometer readings, these data were corrected by reorienting the device to a known heading. Groups of redundant AHRS nodes were successfully averaged in a way where unreliable data from an individual node were ignored.

Data from the AHRSs did not give a consistent time stamp because of latency issues when pinging each AHRS sequentially. To correct for this, the data were time shifted to align all of the AHRSs with the encoders on the laboratory model. This problem will be resolved in the future by moving to a Wi-Fi based system instead of relying on the computer's serial port. The speed increase from a Wi-Fi based system will also reduce the number of instances where an individual AHRS fails to send data. Further works needs to be done with larger networks of redundant nodes such as a full-body motion capture suit to determine the feasibility of using AHRSs for motion capture purposes in real-world settings.

## 4.7 References

[1] Rehabilitation Research and Training Center on Disability Statistics and Demographics, “Annual disability statistics compendium,” 2010.

[2] S. Patel, K. Lorincz, R. Hughes, N. Huggins, J. Growdon, D. Standaert, M. Akay, J. Dy, M. Welsh, and P. Bonato, “Monitoring motor fluctuations in patients with parkinson’s disease using wearable sensors,” in *IEEE Trans. Inf. Tech. Biomed.*, vol. 13, November 2009, pp. 864–873.

[3] T.-T. Nguyen, M.-C. Cho, and T.-S. Lee, “Automatic fall detection using wearable biomedical signal measurement terminal,” in *31st Ann. Int. Conf. of the IEEE EMBS*. IEEE, September 2009, pp. 5203–5206.

[4] M. Kishimoto, T. Yoshida, T. Hayasaka, D. Mori, Y. Imai, N. Matsuki, T. Ishikawa, and T. Yamaguchi, “An internet-based wearable watch-over system for elderly and disabled utilizing emg and accelerometer,” in *Technol. Health Care*, vol. 17, 2009, pp. 121–131.

[5] M. Hanlon and R. Anderson, “Real-time gait event detection using wearable sensors,” in *Gait Pos.*, vol. 30, November 2009, pp. 523–527.

[6] W. S. Yeoh, I. Pek, Y. H. Yong, X. Chen, and A. B. Waluyo, “Ambulatory monitoring of human posture and walking speed using wearable accelerometer sensors,” in *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, 2008, pp. 5184–5187.

[7] M. V. Wieringen and J. Eklund, “Real-time signal processing of accelerometer data for wearable medical patient monitoring devices,” in *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, 2008, pp. 2397–2400.

[8] M. N. Nyan, F. E. Tay, and E. Murugasu, “A wearable system for pre-impact fall detection,” in *J. Biomech.*, vol. 41, December 2008, pp. 3475–3481.

[9] P. Bonato, D. M. Sherrill, D. G. Standaert, S. S. Salles, and M. Akay, “Data mining techniques to detect motor fluctuations in parkinson’s disease,” in *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, vol. 7, 2004, pp. 4766–4769.

[10] B. R. Chen, S. Patel, T. Buckley, R. Rednic, D. J. McClure, L. Shih, D. Tarsy, M. Welsh, and P. Bonato, “A web-based system for home monitoring of patients with parkinson’s disease using wearable sensors,” in *IEEE Trans. Biomed. Eng.*, vol. 58, March 2010, pp. 831–836.

[11] E. D. de Bruin, A. Hartmann, D. Uebelhart, K. Murer, and W. Zijlstra, “Wearable systems for monitoring mobility-related activities in older people: a systematic review,” in *Clin. Reh.*, vol. 22, October 2008, pp. 878–895.

[12] B. Patritti, M. Sicari, L. Deming, F. Romaguera, M. Pelliccio, M. G. Benedetti, D. Nimec, and P. Bonato, “Enhancing robotic gait training via augmented feedback,” in *Eng. Med. Bio. Soc. (EMBC)*, 2010, pp. 2271–2274.

[13] A. Zijlstra, M. Mancini, L. Chiari, and W. Zijlstra, “Biofeedback for training balance and mobility tasks in older populations: a systematic review,” in *J Neu. Reh.*, vol. 7, 2010, p. 15.

[14] Shimmer Research. [Online]. Available: http://www.shimmer-research.com/

[15] Texas Instruments. (2013, February). [Online]. Available: http://www.ti.com/

[16] T. Taylor, S. Ko, C. Mastrangelo, and S. J. M. Bamberg, “Forward kinematics using imu on-body sensor network for mobile analysis of human kinematics,” in *Eng. Med. Bio. Soc. (EMBC)*, 2013.

[17] W. Premerlani and P. Bizard, “Direction cosine matrix imu: Theory,” May 2009, http://gentlenav.googlecode.com/files/MahonyPapers.zip.

[18] (2013, October). [Online]. Available: https://github.com/ptrbrtz/razor-9dof-ahrs/wiki/Tutorial

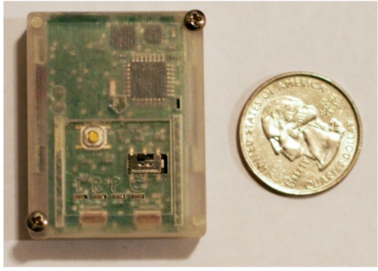

**Figure 4.1**: AHRS sensor with attached XBee$^{TM}$wireless transmitter.

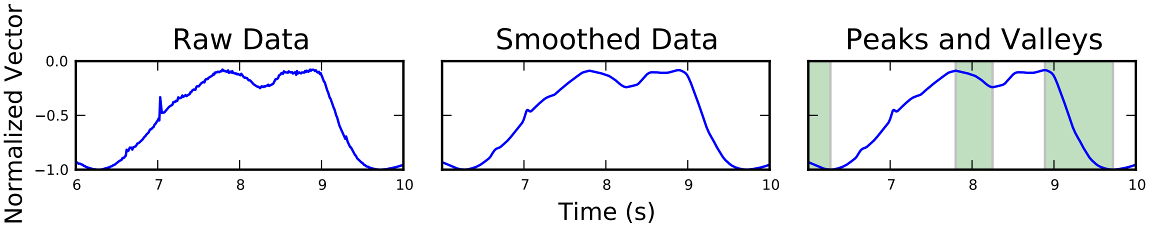

**Figure 4.2**: Comparisons between raw encoder data, encoder data smoothed by 1D 2$^{nd}$-order Gaussian filter, and results from breaking the curve into multiple monotonic curves by locating peaks and valleys. Data were taken from the Foot *forward* X curve.

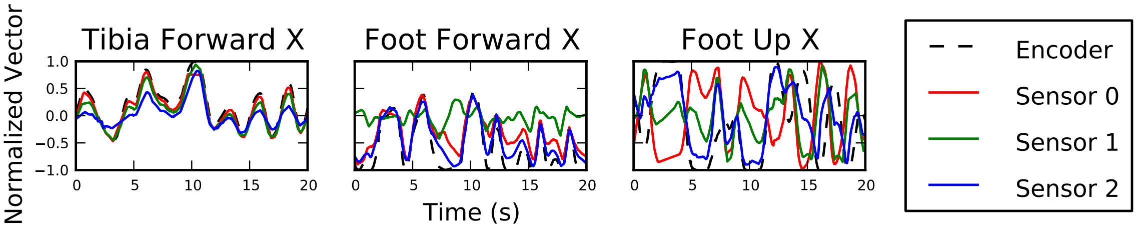

**Figure 4.3**: Raw X-component data for vectors calculated from the AHRS. Note that although there are only three sensors per joint (Tibia and Foot), the foot has enough degrees of freedom to require both a *forward* vector and an *up* vector to be calculated.

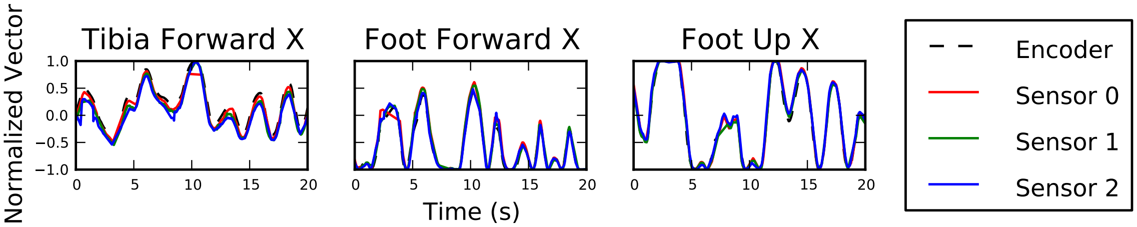

**Figure 4.4**: Output from the calibrated sensors in the *forward* X direction. The dashed curve represents the true value calculated from the mechanical leg's encoders. The red, green, and blue curves represent calibrated output from the three AHRSs.

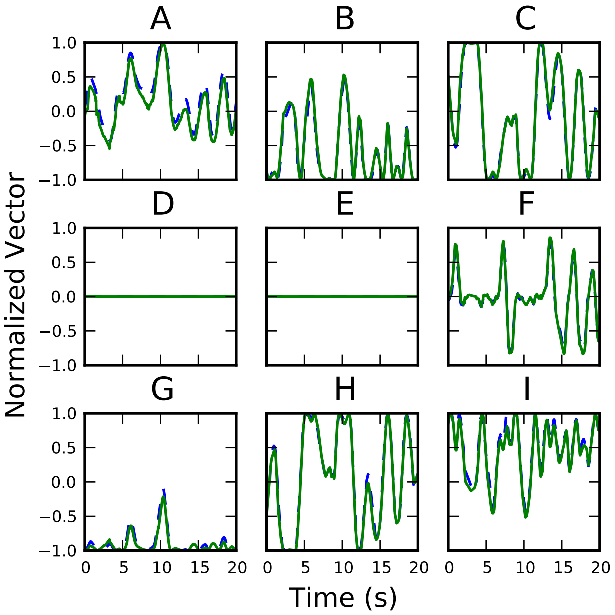

**Figure 4.5**: Fully calibrated and averaged data for all joints and all directions. See **Table 4.1** for corresponding plot titles. The plots shown in *Tibia Forward Y* (D) and *Foot Forward Y* (E) are perfectly flat because motion of the leg was restricted to movement in the X-Z plane. The AHRSs were calibrated by rotating the output to fall inline with this plane, thus these directions have a perfect correlation.

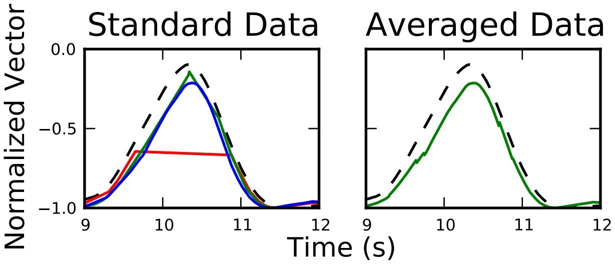

**Figure 4.6**: Comparisons of calibration and averaging methods for the shin joint *forward* Z vector. The left image shows the calibrated nonaveraged data and the right image shows the final data after the averaging algorithm is applied. The slight spikes in the averaged data show points where a sensor either went offline or came back online.

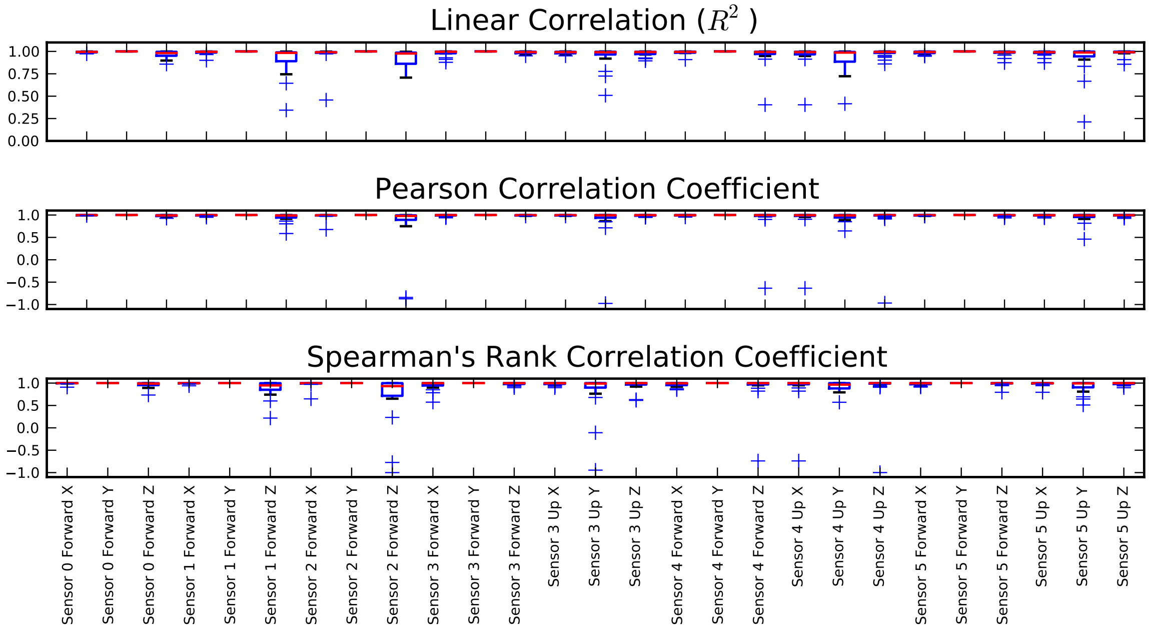

**Figure 4.7**: Box plots showing the correlation coefficients using least-squares ($R^2$), Pearson's product-moment correlation, and Spearman's rank correlation. Sensors 0-2 were attached to the tibia and sensors 3-5 were attached to the foot. For each box, the red line represents the median of the data, the edges of the box represent the 25th and 75th percentiles, the whiskers represent the range of the data not considered outliers, and points outside the whiskers represent outliers. For each correlation, a value of 1 represents perfect correlation.

**Table 4.1**: Averaged values for coefficients over all monotonic sections.

| Direction | $R^2$ | Pearson | Spearman's |
|---|---|---|---|
| A) Tibia Forward X | 0.972 | 0.985 | 0.996 |
| B) Tibia Forward Y | 1.000 | 1.000 | 1.000 |
| C) Tibia Forward Z | 0.954 | 0.976 | 0.917 |
| D) Foot Forward X | 0.992 | 0.996 | 0.973 |
| E) Foot Forward Y | 1.000 | 1.000 | 1.000 |
| F) Foot Forward Z | 0.988 | 0.994 | 0.985 |
| G) Foot Up X | 0.987 | 0.993 | 0.986 |
| H) Foot Up Y | 0.963 | 0.884 | 0.852 |
| I) Foot Up Z | 0.988 | 0.994 | 0.987 |

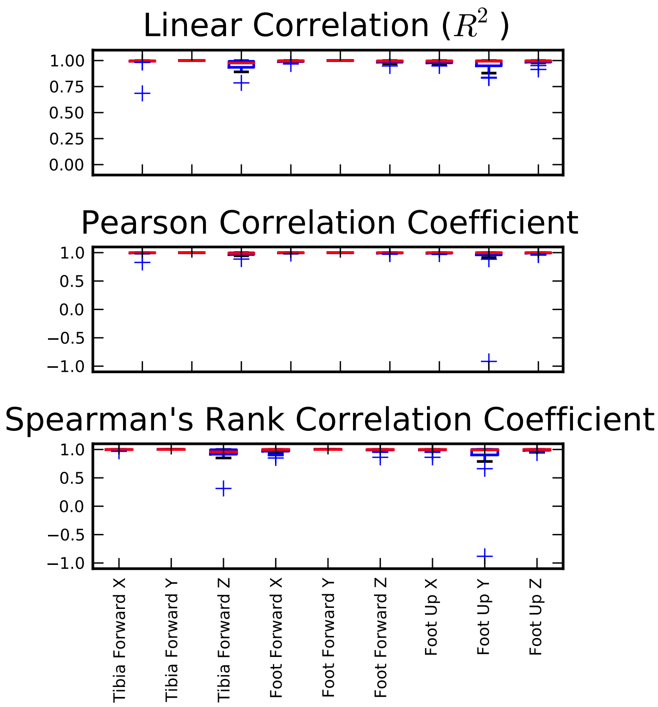

**Figure 4.8**: Box plots showing the correlation coefficients using least-squares ($R^2$), Pearson's product-moment correlation, and Spearman's rank correlation. All of the sensors were averaged using the method defined in this paper. For each box, the red line represents the median of the data, the edges of the box represent the 25th and 75th percentiles, the whiskers represent the range of the data not considered outliers, and points outside the whiskers represent outliers. For each correlation, 1.0 represents perfect correlation.

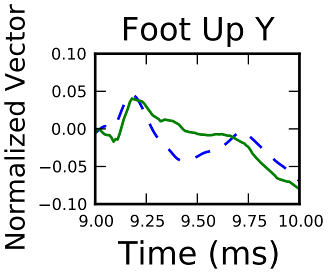

**Figure 4.9**: Section of the Foot Up Y plot showing the period of significant error. Both the Pearson and Spearman's coefficient give a negative value because of the reversed correlation between approximately 10.4 and 10.7-s.

# CHAPTER 5

# POSE-BASED LOCALIZATION ALGORITHM FOR ARBITRARY NODE PLACEMENT

A consumer-grade motion capture system should have an intuitive setup and initialization. For this reason, a localization algorithm is proposed that allows for arbitrary node placement.

## 5.1 Introduction

An attitude heading reference system (AHRS) is a device that fuses an array of sensors to output a global orientation [1]. A collection of AHRS units have been attached to body segments for purposes of motion capture [2], [3]. These systems take the global orientation provided by each AHRS and apply them to the body segments to compute the forward kinematics of the body. By rotating the body segments, the pose of the user wearing the AHRS nodes is determined.

This motion capture system requires a foreknowledge of the placement of all attached AHRS modules so that the correct global orientation is applied to each body segment. This requires the user to either manually enter the location of each AHRS unit or carefully place each AHRS on the body according to some predetermined map.

To simplify this initialization process, a method to *localize* each node on the body using by mimicking a series of reference poses and comparing the global orientation output of each ARHS module. This allows a user to arbitrarily place each AHRS node and then initialize the system without having to manually input each AHRS location.

Previous work in finding the location of on-body sensors has been accomplished by analyzing the output of accelerometers while the wearer performs everyday motions such as walking [4]–[7]. These methods require a significant amount of data capture to properly localize the sensors and limit the set of possible locations. For example, the applications these localization routines target do not require the left or right side of the body to be determined. A rotation-based localization method was implemented with mobile devices with the possible locations limited to pockets [8]. Therefore, for purposes of full-body motion capture, these routines are not sufficient.

Localization methods such as particle filters have been used in conjunction with inertial measurement units [9] and sonar [10] to estimate the location of a person or vehicle in a known space. Similar concepts can be applied to on-body AHRS localization by treating all of the possible bone segments as a "map."

A novel approach to localizing an array of AHRS nodes to body-segment locations is proposed. This approach uses a set of reference poses that are mimicked by a user in order to compare the global orientations of the AHRS nodes with the global orientations of the body segments of the reference pose. Further, a minimum number of poses necessary to localize a full suite of AHRS modules is investigated.

## 5.2 Background

To implement the localization algorithm, hereby referred to as the *Localizer*, a group of reference poses were created that isolated possible body locations. Quaternions were used to evaluate and compare the global orientations of each AHRS with the global orientations of the locations in the reference poses. A pseudo-probability framework was designed to normalize quaternion differences of all AHRS nodes against each location. This framework was designed so that it could be multiplied across poses which forces the AHRS nodes to converge on a location.

To evaluate the effectiveness of the Localizer, naïve subjects were recruited. They attached an AHRS to each possible locations on their body and mimicked all reference poses. For each subject the locations of the AHRS nodes and the order of the reference poses were randomized.

Reference poses needed to be clear to the subjects so that they could properly mimic each pose. Because AHRS motion capture needs a reference position to base all forward kinematics on, poses that move the center of mass relative to the ground need to be counteracted. A method to determine the center of mass based on the user's height and gender was created. In order to make following the poses intuitive to the subjects while viewing them on a monitor, a method to mirror the poses about the global Y-axis was developed.

### 5.2.1 Particle Filter

A particle filter is a technique that has been used to localize sensors and robots. Ascher *et al.* used a particle filter to create a map-matching algorithm for low-cost IMUs [9]. A particle filter using sonar readings was used to locate the position of an autonomous underwater vehicle [10]. These methods can be extended to on-body localization by transforming the joints of the human body into a list of possible locations.

Particle filters work by generating "particles" that represent possible locations on a known map. For each particle that is generated, an estimate for the sensor readings that would be observed at that location are calculated. The estimated sensor readings of each particle are compared with the actual sensor readings. This process is then repeated with a new set of particles that are weighted to occur closer to the "most probable" locations of the previous sampling. As this cycle continues, the particles begin to conglomerate around the most probable location on the map.

A particle filter is basically a Bayes filter that is created by a set of samples instead of a general probability distribution.

$$p(x) \approx \frac{1}{N} \sum_i \delta_{x(i)}(x)$$

By reducing the probabilities to a fixed number of particles, no probability distribution is necessary.

The location is further refined by having the entity to be localized (typically a mobile robot) explore the space and compare its readings against the map. As the robot continues to move around, the particles will pack around the most likely location.

### 5.2.2 Grid-Based Localization

Grid-based localization is similar to a particle filter, but the map is broken down into a grid that limits the number of possible locations [11]. The computation needs of the localization procedure decreases with large grid elements. Essentially, the "resolution" of the map determines the computational needs of the algorithm.

This method differs from a particle filter based approach by having a fixed number of possible locations. The estimated sensor readings of each element on the grid can be preprocessed so that the only necessary computation is the comparison between the estimated and actual sensor readings. Each grid tile is assigned a probabilty for the likeliness that the entity to be localized is present in it. Again, as the robot roams the map and takes sensor readings, the probabilities of the grid adjust to localize the robot.

Grid-based localization is based on the Bayesian inversion formula:

$$p(x \mid s_1 \wedge \ldots s_{n-1} \wedge s_n \wedge m) = \frac{p(s_1 \wedge \cdots \wedge s_{n-1} \wedge s_n \mid x \wedge m) \cdot p(x \mid m)}{p(s_q \wedge \cdots \wedge s_{n-1} \wedge s_n \mid m)}$$

Both particle filters and grid-based localization algorithms rely on a static map of the environment and the robot to roam the environment and take sensor readings. With the localization of AHRS units attached to the body, the order is reversed. The "robot" (AHRS nodes) stay stationary and the "map" (the segments of the body) move to different positions. The underlying principle is the same - multipe readings are taken from different positions and compared to estimated sensor readings at each location. In order to get the location probabilities to converge more than one full-body pose is necessary.

### 5.2.3 Quaternion convention

A quaternion is a specialized four-dimensional vector that can be used to describe an orientation in three-dimensional space. The four components of the vector represent the three components of a axis of rotation and the angle of rotation about that axis. There is no single convention for portraying quaternions and equations involving quaternions. For the purposes of this paper, the quaternion convention used will be described below.

A quaternion is represented as a bold-faced letter such as $\mathbf{q}$. When a quaternion is preceded by a subscript and superscript, this will reference the initial and final coordinate frame after the rotation is applied. As an example, ${}^{B}_{A}\mathbf{q}$ represents a rotation from coordinate frame $A$ to coordinate frame $B$. The quaternion parameters are represented as a four dimensional vector:

$$\mathbf{q} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

where $w$ represents the angle of rotation about the axis described by $x$, $y$, and $z$. In order for quaternions to be valid, they must be normalized across these four values. $\mathbf{q}^{-1}$ represents the inverse of $\mathbf{q}$ and can be found by negating the $x$, $y$ and $z$ values of the quaternion. This inverse of a quaternion is equal to the quaternion with the two coordinate frames reversed.

$${}^{B}_{A}\mathbf{q}^{-1} = {}^{A}_{B}\mathbf{q}$$

A quaternion product is represented by the symbol $\otimes$. It should be noted that the order of quaternion multiplication is important as quaternion multiplication is no commutative. In other words, $\mathbf{q}_1 \otimes \mathbf{q}_2 \neq \mathbf{q}_2 \otimes \mathbf{q}_1$.

Multiplying two quaternions compound the two rotations and can be chained to move from one major coordinate frame another via a series of sub-coordinate frames:

$$ {}^{D}_{A}\mathbf{q} = {}^{B}_{A}\mathbf{q} \otimes {}^{C}_{B}\mathbf{q} \otimes {}^{D}_{C}\mathbf{q} \tag{5.1}$$

By summing the multiplied values of similar components of two quaternions, the quaternion dot product is found:

$$\mathbf{q}_1 \cdot \mathbf{q}_2 = q_{0,w}q_{1,w} + q_{0,x}q_{1,x} + q_{0,y}q_{1,y} + q_{0,z}q_{1,z} \tag{5.2}$$

To rotate a vector by a quaternion, the following can be used [12]:

$$ {}^{B}\mathbf{v} = {}^{A}_{B}\mathbf{q} \otimes {}^{A}\mathbf{v} \otimes {}^{A}_{B}\mathbf{q}^{-1} \tag{5.3}$$

### 5.2.4 Probability Maps

A framework to represent the relative likelihood of an AHRS to be located on one body segment as opposed to another needed to be designed. The concept of a *Probability Map* that collects a cross-reference of pseudo-probabilities into a single object will be described below.

A Probability Map is a two-dimensional array of values that gives a pseudo-probability of any AHRS being located at any of the possible body segment locations. These pseudo-probabilities are generated by finding the "error" between the global orientation of the reference location and the global orientation of an AHRS module.

The rotational difference between two quaternions is found with:

$$error = \frac{\arccos(2(\mathbf{q}_1 \cdot \mathbf{q}_2)^2 - 1)}{\pi} \tag{5.4}$$

Equation (5.4) returns a number between 0 and 1 representing the rotational distance between two quaternions. This distance can be difficult to visualize, but it is rotational distance between two orientations like there is a distance between two points in three-dimensional space. A value of 1 means the two quaternions are a maximum distance between each other, in other words, they are inverses of each other. A value of 0 means the two quaternions are the same.

If a faster computation of (5.4) is needed, a rough approximation is found using

$$error_{approx} = 1 - (\mathbf{q}_1 \cdot \mathbf{q}_2)^2 \tag{5.5}$$

The Probability Map consists of rows of quaternion errors that are normalized across all sensors with respect to a single location. Because the rows are normalized, they can be treated like a probability distribution and be multiplied across multiple reference poses as long as the output is also normalized along its rows. These rows **are not** real probabilities; they are pseudo-probabilities. However, for the purposes of this paper, they will be referred to as probabilities.

### 5.2.5 Subject recruitment

To validate that the algorithm works for people unfamiliar with the system, twelve subjects were recruited. The proposed study was reviewed by a University of Utah staffer who, with her colleagues, agreed that because this study did not propose to collect information about the participants in a manner that satisfied the federal definition of a ″human subject,″ this study did not require Institutional Review Board (IRB) approval.

### 5.2.6 Reference poses

The Localizer depends on comparing the global orientation of AHRS nodes to reference global orientations. For the purposes of AHRS motion capture, the human body was simplified as eleven body segments, namely:

1. Head
2. Chest
3. Stomach
4. Right upper arm
5. Right lower arm
6. Left upper arm
7. Left lower arm
8. Right upper leg

9. Right lower leg
10. Left upper leg
11. Left lower leg

The Localizer requires a reference rotation for each of these segments. In order for the Localizer to make a proper comparison, reference poses were generated by attaching AHRS nodes on all eleven segments and recording their orientations.

### 5.2.7 Pose mirroring

The Localizer needs users to mimic reference poses. To make this process intuitive, the reference poses are displayed in front of the user so that the reference pose can be viewed while performing the pose. The poses need to be mirrored about the plane between the user and the screen so that it makes the poses easier to follow. A method to mirror poses based on global orientations was needed.

### 5.2.8 Center of mass calculation

Each AHRS node provides a global orientation that is assigned to one of the body segments of the digital avatar. These body segments are referred to as "bones." The process that displays the graphics requires that the avatar must have a *root* bone that determines the global position and orientation of the avatar as an entire body. All bones are then *children* of the root bone in *parent-child* relationships. The pose of the avatar can be described as a series of vectors connected in a chain; the position of the root bone has vectors that branch out that eventually lead to the fingertips or toes. By knowing the length and global orientation of each vector representing a bone, the location of the end of any vector bone can be determined.

One difficulty that arises is that the root bone's position acts like an anchor for all of the bones down the chain. This means that if the pose is meant to be in a squatting position, the feet of the avatar will appear to float in the air instead of touching the ground. The root bone needs to move with the center of mass of the person wearing the motion capture system.

The center of mass of the user is calculated from the locations of the centers of mass of each equivalent bone of the users body. Even though body types differ from one person to another, an approximation of the centers of mass for each bone can be found from gender and height. Once the center of mass is found, the root bone of the avatar can be translated to match the center of mass of the user.

## 5.3 Methods

The Localizer is an algorithm that uses Probability Maps and combines them across reference poses. Reference poses were chosen in an attempt to strike a balance between adequately isolating the possible locations and covering as many locations as possible in a single pose. The ultimate goal of the Localizer is to properly place every connected AHRS node and do so in as few poses as possible.

To evaluate the Localizer, subjects were outfitted with AHRS modules and mimicked a series of reference poses. Reference poses were displayed to the user with a monitor placed directly in front of each subject. To make the poses as easy to copy as possible, a technique was developed to mirror the poses across the plane that separates the subject from the reference pose. Some poses required the root bone of the reference avatar to be moved to match the center of mass of the avatar relative to the ground. A method to determine the center of mass of a user was created.

The efficacy of the Localizer was determined by applying the algorithm to all of the subjects performing all of the reference poses. Best- and worst-performing subject data sets were identified and investigated for insights. A method to determine the minimum number of poses required to correctly locate as many sensors as possible was deployed to find an “ideal” set of reference poses.

### 5.3.1 Localizer

The Localizer works by taking the global orientations of every body segment of a reference pose and comparing the global orientations of every connect AHRS module in order to build a Probability Map. A Probability Map is generated for every pose and these Probability Maps are multiplied together. When two Probability Maps are multiplied together, the resultant Probability Map is a fusion of the data that causes the correct locations to converge to its true value.

#### 5.3.1.1 Probability Map

A series of reference poses are generated by attaching a set of AHRS nodes in predetermined locations and recording the poses. To localize a set of AHRS nodes, the nodes are recorded while the user mirrors the reference poses. For each reference pose, a probability map of all the attached AHRS modules is created by correlating the output of each module to the global orientation of all possible bone locations.

The localizing algorithm starts by comparing the global orientation of each AHRS unit with the global orientation of a *single* possible bone location. For example, if four AHRS nodes are attached to the user while a reference pose is being mirrored, the global quaternion comparisons might look like **Table 5.1**.

By normalizing the values across a row so that the sum of the row equals 1.0, a pseudo-probability distribution is created for every AHRS node in relation to that one location. This is done by dividing the correlation value of each AHRS by the sum of all the correlation values in the row.

$$normalized_i = \frac{ahrs_i}{\sum\limits_{j=1}^{n} ahrs_j} \tag{5.6}$$

**Table 5.2** shows this normalization.

By making a pseudo-probability for every connected AHRS module in relation to every possible node location, a table of all possible correlations is created. This table will be called a *Probability Map*. **Table 5.3** shows an example four-bone skeleton with four attached AHRS nodes.

Note that the Probability Map is not normalized vertically; if the vertical values are summed, a value of 1.0 is not guaranteed. However, the values in the vertical columns can be used as "probability weights," with the highest value in a given column used to determine the most likely location of a node as shown in **Table 5.4**.

#### 5.3.1.2 Multiple Pose Localization

The benefit of normalizing rows to act like probabilities is that Probability Maps can be multiplied across multiple poses to refine the probabilities. By multiplying the rows of the probability map across multiple poses, the pseudo-probability that a given AHRS node is at a given location begins to converge. As an example, **Table 5.5** shows probability distributions across four poses. This **does not** represent a single Probability Map, but is instead the rows for the same bone across multiple poses.

Probability Maps can be updated with new pose data by multiplying the same elements and storing them in a new Probability Map at the same location. However, it is important to renormalize each of the rows of the new Probability Map so that the sum of the row is still equal to 1.0 and each element is a percentage.

$$temp_{i,j} = (a_{i,j})(b_{i,j}) \qquad \text{Multiply like-elements} \tag{5.7a}$$

$$c_{i,j} = \frac{temp_{i,j}}{\sum\limits_{k=1}^{n} temp_{i,k}} \qquad \text{Normalize row} \tag{5.7b}$$

When (5.7) are applied across all rows of **Table 5.5**, a new Probability Map is formed. **Table 5.6**, **Table 5.7**, and **Table 5.8** show the progression of multiplying similar rows from different poses.

Even though Pose 4 from **Table 5.5** does not heavily favor the correct location like the other poses, **Table 5.8** shows it still contributes toward convergence because of the renormalization procedure. By following (5.7), the probability of each AHRS at a given location will tend to converge toward 0.0 or 1.0.

### 5.3.2 Pose Mirroring

Reference poses are displayed on a screen in front of the user. This requires the screen to act as a mirror, flipping the orientations and bone locations about the vertical plane. Because the output of the AHRS nodes needs to be compared to the reference poses, the reference pose must be mirrored to the user to align and orient the data correctly.

First, the possible bone locations need to be reversed right-for-left for all bones that branch away from the vertical center dividing the body. Bone segments such as the head or waist do not have a reciprocal bone to switch locations, but bones like the right upper arm or right lower leg do.

The inverse of a quaternion is created by negating the three components of the quaternion that represent the axis of rotation. However, an inverse of the quaternion is not the same as mirroring the rotation about the global vertical axis. With the global coordinate frame defined as $Y$ for the vertical component and $Z$ as the forward component or the component the user stares along while in the default position, the $z$ and $w$ components of the reference pose quaternion should be negated. This has the effect of rotating the reference quaternion about its *local* $Z$ axis to flip the bone across the *global* $Y$-$Z$ plane. This quaternion is then rotated around the global $Y$-axis to achieve the mirrored rotation. In practice, this looks like:

$$\mathbf{q}_{mirrored} = \begin{bmatrix} x_{ref} \\ y_{ref} \\ -z_{ref} \\ -w_{ref} \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{5.8}$$

### 5.3.3 Center of Mass Calculation

Poses are constructed by setting the orientations of each avatar bone to the global orientation provided by each AHRS node. The vectors are put in *parent-child* relationships, which means that the tail of a child vector is inherited by the tip of its parent. Because the AHRS nodes provide global instead of local orientations, the vectors of the simplified skeleton do not inherit orientations from the parent vectors.

All global orientations are represented using quaternions. Each vector representing a bone can be viewed as having an identity rotation applied to it. When a global orientation is applied to the vector, the resultant vector is found from

$$^{1}\mathbf{v} = {}^{0}_{1}\mathbf{q} \otimes {}^{0}\mathbf{v} \otimes {}^{0}_{1}\mathbf{q}^{-1} \tag{5.9}$$

where $^{0}\mathbf{v}$ is the original vector, ${}^{0}_{1}\mathbf{q}$ is the global orientation quaternion, and $^{1}\mathbf{v}$ is the new vector.

Vector lengths and center of mass locations are initialized using the values in **Figure 5.1** and **Figure 5.2** and **Table 5.9** and **Table 5.10**. **Figure 5.3** shows the simplified skeleton. The location of the center of mass is given by proximal values which is a percentage of the magnitude of the vector along the vector's direction. This means that the global center of mass location is found by

$$\mathbf{v}_{com} = \mathbf{v}_{tip,parent} + p\left({}_{bone}^{0}\mathbf{q} \otimes {}^{0}\mathbf{v}_{bone} \otimes {}_{bone}^{0}\mathbf{q}^{-1}\right) \tag{5.10}$$

where $p$ is the proximal percentage of the length and ${}_{bone}^{0}\mathbf{q}$ is the global quaternion of the bone in question. The center of mass of the user can now be found from

$$\mathbf{v}_{COM} = \sum_{i=1}^{n} m_i \mathbf{v}_{com,i} \tag{5.11}$$

where $m_i$ is the percent mass of the current bone, $\mathbf{v}_{com,i}$ is the global position of the current bone's center of mass, and $\mathbf{v}_{COM}$ is the center of mass of the entire body.

Calculation of the overall center of mass does not take into account the distance from the center of mass and the floor. To add this component, only the vertical component of the global coordinate frame is considered. The minimum vertical component from the list of vector bone *tips* is the distance to move the avatar relative to the floor. Applying both the vertical floor distance and the center of mass calculation only works for *static* poses. If the user leaves the ground, the minimum vertical component is no longer the global distance to the floor and other position tracking methods must be used.

### 5.3.4 Reference poses

The twenty-four reference poses designed to test the Localizer are shown in **Figure 5.4**. All poses were selected to isolate specific body segments.

### 5.3.5 Localizer efficacy validation

To validate that the algorithm works for people unfamiliar with the system, twelve subjects were invited to test the system by following a series of reference poses while wearing the AHRS motion capture system. Each pose was created with the intent of isolating a collection of bones by orientating them in different directions; the greater the disparity between rotations between bones, the better the localization algorithm performs. Every pose was designed to have the head facing forward so any subject can actively look at the reference pose while performing the pose.

For each subject, eleven independent AHRS nodes were placed on each of the following locations:

1. Head

2. Chest

3. Stomach

4. Right upper arm

5. Right lower arm

6. Left upper arm
7. Left lower arm
8. Right upper leg
9. Right lower leg
10. Left upper leg
11. Left lower leg

The location of each AHRS node was randomized for each subject with the constraint that a single AHRS node was present on each possible body segment.

To initialize the system, the subject was told to stand at a designated spot of the room and facing the same direction. The subjects stood in this spot with their legs spread shoulder-width apart, arms straight to the side and facing forward. This first pose is the default initialization pose that allows the system to find the default orientation of each attached AHRS node. This is necessary because placement of the nodes can variate between subjects.

A computer monitor displaying a mirrored reference pose was placed in front of the subject. The subject was told to mirror the pose they saw on the monitor and that each pose was designed to be looked at while performing the pose. Each subject was informed that they would follow a series of twenty-four poses and that they should hold each pose until a new pose appeared on the screen.

The order of the twenty-four poses was randomized for each subject. A second computer monitor only available to the administrator displayed the reference pose as well as the real-time output of the AHRS nodes attached to the subject. This allowed the administrator to ensure that all hardware was working properly before recording the pose of the subject. The subject's pose was recorded for every reference pose.

After data capture, Probability Maps for each individual pose was generated. The max probability weight in each column of the Probability Map was used to pick a location for each attached AHRS node. The chosen locations were compared to the list of true locations.

### 5.3.6 Pose characterization

**Figure 5.5** shows what body segments are targeted by each reference pose. A circle placed in a pose column represents a body segment that has a global orientation that is significantly different from the other body segments.

Some poses are designed to target a small number of segments whereas others are designed a high number. **Table 5.11** shows the number of body segments targeted for each pose. The grid location of each number corresponds to the location of the pose in **Figure 5.4**. The numbers in this table are found by counting up the number of circles for a pose in **Figure 5.5**.

The number of circles in the rows of **Figure 5.5** show how many poses each body segment is targeted in. For the Localizer to be effective, each body segment needs to be well represented by the set of reference poses used. **Table 5.12** displays the number of poses that target each body segment.

### 5.3.7 Subject characterization

The best- and worst-performing subjects were determined by analyzing the probabilities of each sensor at its correct location after being localized across all reference poses. The probability of the correct location of each node was compiled in a list for every subject. By multiplying all of the values in this list, the overall rank of each subject was determined. Higher probability values for each sensor produces a high rank and low probabilities produce a low rank. Therefore, the subject with the lowest overall rank represents the subject that was the “worst” at mimicking the reference poses and the subject with the highest overall rank represents the subject that was the “best” at mimicking the reference poses.

### 5.3.8 Minimum number of poses for localization

Using the twelve subjects, the minimum number of poses from the set of twenty-four that will properly localize all of the attached nodes was determined. It is not the individual poses that lead to a proper localization, but the combination of a set of poses. Out of twenty-four poses, the number of possible combinations is large enough that special measures have been employed to arrive at a solution early.

A constraint placed on the minimum number of poses was that the list needs to correctly identify the location of each AHRS node. Therefore, when looping through the subjects with a candidate pose set, as soon as one subject fails to be properly localized, a new candidate set is selected. Further, the first pose set that correctly localizes all of the attached nodes on all twelve subjects is chosen as the minimum set and the search is ended.

Candidate sets were determined using CombinatronicsLib ver. 2.1. CombinatronicsLib is an open source Java library providing support for generating unique subsets for a list of objects.

## 5.4 Results

Probability Maps are represented as an 11 × 11 heat map grid of the probability of each ARHS node with respect to each possible location. With twenty-four poses for each subject, Probability Maps are presented as a 6 × 4 array of Maps. The location of these Maps correspond with the locations of the poses presented in **Figure 5.4**. Even though each subject had a randomized location for each AHRS node, all Probability Maps are arranged so that the correct locations of each node fall along the diagonal.

**Table 5.13** shows the final Localizer output for all subjects for all possible locations. A match in the column for a subject means the AHRS at that location was correctly localized.

A comparison between the pose characterization and the rotation disparities from identity are given in **Figure 5.6**. Cells that are shaded darker represent body segments that are moved significantly from the base pose. **Figure 5.7** shows an example of how each Probability Map will be displayed.

Two types of Probability Maps are presented. One shows the Probability Map generated for a single reference pose with respect to the subject's matching pose. This will be designated as an *individual* Probability Map. The other Probability Map type shows the progression of the localizing algorithm as the Probability Maps are multiplied together. This will be designated as a *progression* Probability Map. Individual Probability Maps do not have a large contrast between elements; the color scaling on these maps is scaled to the maximum and minimum of the maps for easier viewing. Progression Probability Maps are scaled to the full range of 0.0-1.0.

The overall ranking of each subject are presented in **Table 5.14** with the worst- and best-performing subjects highlighted.

Two subjects were selected for discussion and their Probability Maps are presented in **Figure 5.8**, **Figure 5.9**, **Figure 5.10**, and **Figure 5.11**. These two subjects represent the worst- and best-performing data sets found in **Table 5.14**. The Probability Maps for all subjects are provided as an appendix.

The minimum number of poses in order to successfully locate every node for every subject was found to be eight. The set of eight poses first identified by the minimization are shown in **Figure 5.12**.

## 5.5 Discussion

The Localizer successfully located every AHRS when all twenty-four Probability Maps were multiplied together (**Table 5.13**). Every AHRS node was successfully localized across all twelve subjects.

### 5.5.1 Probability Maps

The Probability Maps shown in **Figure 5.8** and **Figure 5.10** indicate how certain poses target certain bones based on larger disparity from the neutral standing pose. For example, $\text{Pose}_{2,2}$ in **Figure 5.8** corresponds with a pose that isolates the head (**Figure 5.13**). This particular pose produces a Probability Map where the majority of cells are uncertain. Row 1 of this map represents the head with column 1 being the corresponding AHRS node. It is important to note that the default pose is standing with feet at shoulder width and arms to the side. This means that if the user stands in this pose, all of the bones have a global orientation that is the identity quaternion.

The effect of this can be seen in **Figure 5.14** where all of the elements have roughly the same probability except for the first row and first column. $\text{Pose}_{2,3}$ has only one bone that deviates from the base identity quaternion - the head. Other than the head and its corresponding AHRS node, all of bones and nodes have the global orientation. When this is normalized across rows, it causes the distribution to be flat across similar sensors with a peak in column 1, row 1 or a valley as in column 1, rows 2-11.

### 5.5.2 Location selection

The importance of locating each AHRS node by finding the maximum probability along that node's column is demonstrated by **Figure 5.15**. One location that is particularly hard to separate from the others is the stomach location. This is because the chest location is connected to the stomach location and it is difficult to design a pose that forces the stomach and chest to have a large difference between their global orientations. In **Figure 5.15**, the chest is represented by row 2 and the stomach is represented by row 3. Row 3 incorrectly places the sensor in column 2 as the most likely sensor attached to the stomach. If only rows are considered, the localization algorithm concludes that the sensor in column 2 is attached to both the chest and stomach which is not possible.

When the individual sensors are localized by columns, this problem of a single sensor attached to multiple locations disappears. The localization algorithm correctly selects the sensor in column 2 to be the sensor connected to the chest (row 2) and the sensor in column 3 as the sensor connected to the stomach (row 3).

### 5.5.3 Robustness

The localizing algorithm converges on the correct solution even when the input data is not ideal. Subject 3 struggled to perfectly match the poses. **Figure 5.16**, **Figure 5.17**, and **Figure 5.18** show one of the extreme cases where the errors between the reference pose and actual data capture from subject 3 are readily apparent.

### 5.5.4 Pose efficacy

Poses that are oriented to the side instead of facing forward proved to be difficult for the subjects to emulate. $\text{Pose}_{5,1}$, $\text{Pose}_{5,2}$, $\text{Pose}_{6,3}$, and $\text{Pose}_{6,4}$ frustrated the subjects because it was not immediately clear which arm or leg should be facing the display. **Figure 5.19** shows these four poses. The other twenty poses were intuitive for the subjects to mirror.

A minimum of eight poses were required to properly locate every attached AHRS node for each of the twelve subjects. A set of poses that produced this result were $\text{Pose}_{1,1}$, $\text{Pose}_{1,3}$, $\text{Pose}_{1,4}$, $\text{Pose}_{2,1}$, $\text{Pose}_{2,4}$, $\text{Pose}_{4,2}$, $\text{Pose}_{4,3}$, and $\text{Pose}_{5,3}$. These poses are presented in **Figure 5.12** as a set. Four of the eight poses move the chest off center. Considering the difficulty the localizing algorithm has with separating the chest location from the stomach location, it is likely that more data is needed to properly place an AHRS node attached to this body segment. Two knee-raise poses localize nodes attached the legs and the head orientation is different enough from the chest in enough of the poses to localize this segment.

## 5.6 Conclusion

The accurate localization of all sensors for every subject showed the effectiveness of the proposed localization algorithm. Probability Maps that are multiplied across successive poses can be used to identify a solution for the location of an attached AHRS module. The minimum of eight poses from the proposed set of twenty-four allows the user of an AHRS motion capture system to properly place attached AHRS nodes without the need for manual entry. Further, the given set of eight poses contains poses that all face the user directly and avoids poses that confuse the user.

The localization algorithm proposed in this paper uses static poses to build each Probability Map. Every localization pose was captured one-at-a-time; a method of quickly transitioning from one pose to another could further improve the length of time to properly set up a motion capture system. Because each pose only provides one corresponding Probability Map, a method that continues to localize the sensors while the user is transitioning from one pose to another could provide more data to either make localizations quicker or more robust. Additional poses could be required if a minimum certainty has not been met.

As mobile motion capture systems based on wireless AHRS nodes become more prevalent, a quick and intuitive method such as described here will be a necessary component for initializing the system for use in consumer-friendly environments. The localization algorithm presented in this paper provides a way to make affordable and portable motion capture available to a wider market.

## 5.7 References

[1] A. M. Sabatini, "Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing," *Bio. Eng., IEEE Trans.*, vol. 53, no. 7, pp. 1346–1356, 2006.

[2] T. Taylor, S. Ko, C. Mastrangelo, and S. Bamberg, "Forward kinematics using imu on-body sensor network for mobile analysis of human kinematics." in *IEEE Eng. Med. Bio. Soc. Conf.*, vol. 2013, 2013, pp. 1230–1233.

[3] M. Di Capua and D. Akin, "Body pose measurement system (bpms): An inertial

motion capture system for biomechanics analysis and robot control from within a pressure suit," in *Int. Conf. Env. Sys.: AIAA*, 2012.

[4] N. Amini, M. Sarrafzadeh, A. Vahdatpour, and W. Xu, "Accelerometer-based on-body sensor localization for health and medical monitoring applications," *Perv. Mob. Comp.*, vol. 7, no. 6, pp. 746–760, 2011.

[5] K. Kunze, P. Lukowicz, H. Junker, and G. Tröster, "Where am i: Recognizing on-body positions of wearable sensors," in *Loc. Cont. Aw.* Springer, 2005, pp. 264–275.

[6] A. Vahdatpour, N. Amini, and M. Sarrafzadeh, "On-body device localization for health and medical monitoring applications," in *Perv. Comp. Com. (PerCom), 2011 IEEE Int. Conf.* IEEE, 2011, pp. 37–44.

[7] W. Xu, M. Zhang, A. A. Sawchuk, and M. Sarrafzadeh, "Robust human activity and sensor location corecognition via sparse signal representation," *Bio. Eng., IEEE Trans.*, vol. 59, no. 11, pp. 3169–3176, 2012.

[8] Y. Shi, Y. Shi, and J. Liu, "A rotation based method for detecting on-body positions of mobile devices," in *Proc. 13th Int. Conf. Ubi. Comp.* ACM, 2011, pp. 559–560.

[9] C. Ascher, C. Kessler, M. Wankerl, and G. Trommer, "Dual imu indoor navigation with particle filter based map-matching on a smartphone," in *Ind. Pos. Ind. Nav. (IPIN), 2010 Int. Conf.* IEEE, 2010, pp. 1–5.

[10] F. Maurelli, S. Krupinski, Y. Petillot, and J. Salvi, "A particle filter approach for auv localization," in *Oceans 2008 Conf. Proc.* IEEE, 2008, pp. 1–7.

[11] W. Burgard, D. Fox, D. Hennig, and T. Schmidt, "Estimating the absolute position of a mobile robot using position probability grids," in *Proc. Nat. Conf. Art. Intel.*, 1996, pp. 896–901.

[12] J. B. Kuipers *et al.*, *Quaternions and Rotation Sequences,* Vol 66 Princeton, NJ, USA: Princeton UP, 1999.

[13] R. Drillis, R. Contini, and M. Bluestein, "Body segment parameters; a survey of measurement techniques." *Art. Limbs*, vol. 8, pp. 44–66, 1963.

[14] R. Huston, *Principles of Biomechanics,* Boca Raton, FL, USA: CRC Press, 2008.

[15] D. A. Winter, *Biomechanics and Motor Control of Human Movement,* New York, NY, USA: John Wiley & Sons, 2009.

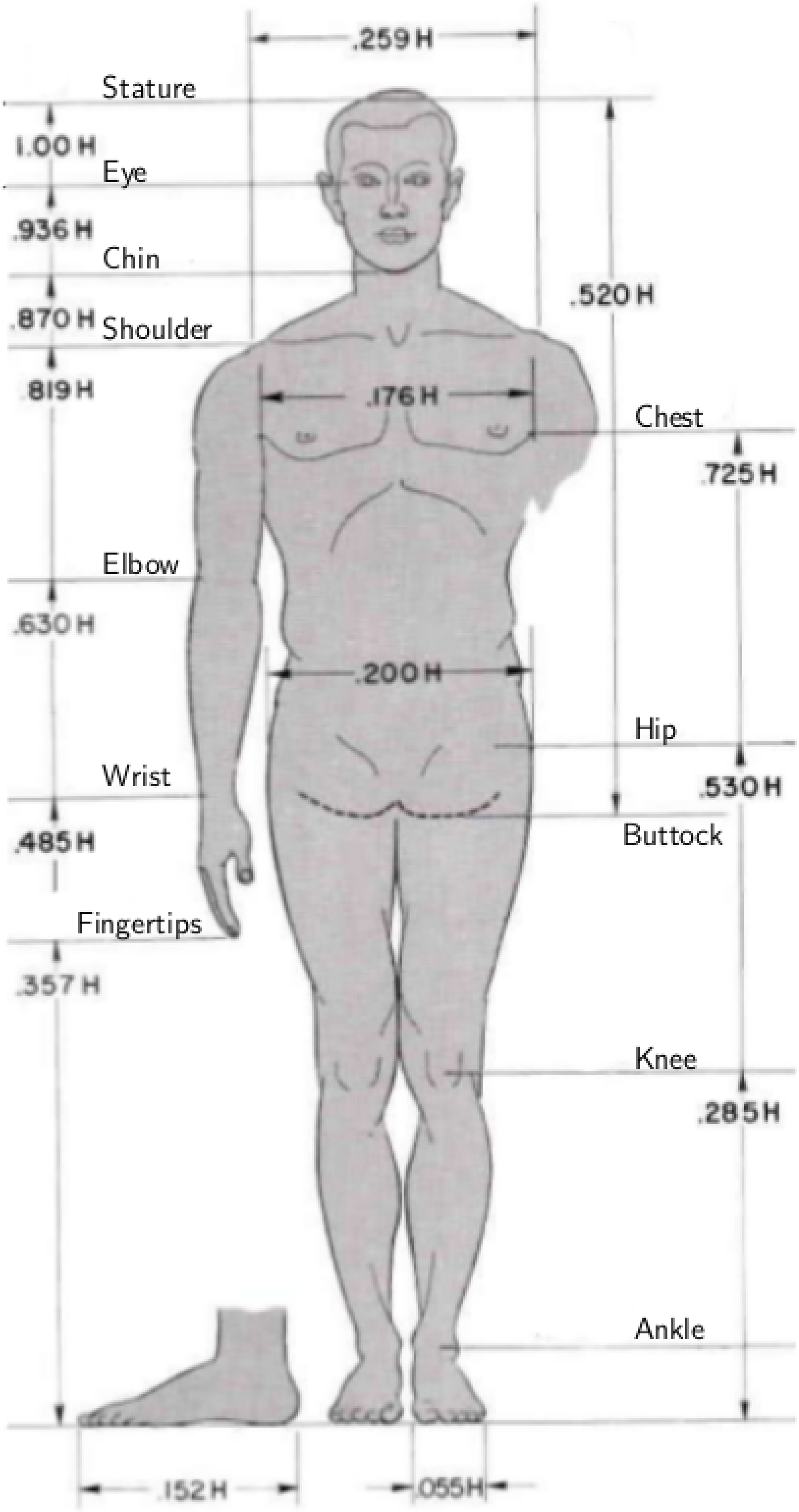

**Figure 5.1**: Male body segment lengths relative to user height. These values are used to calculate the vector lengths for the simplified skeleton [13].

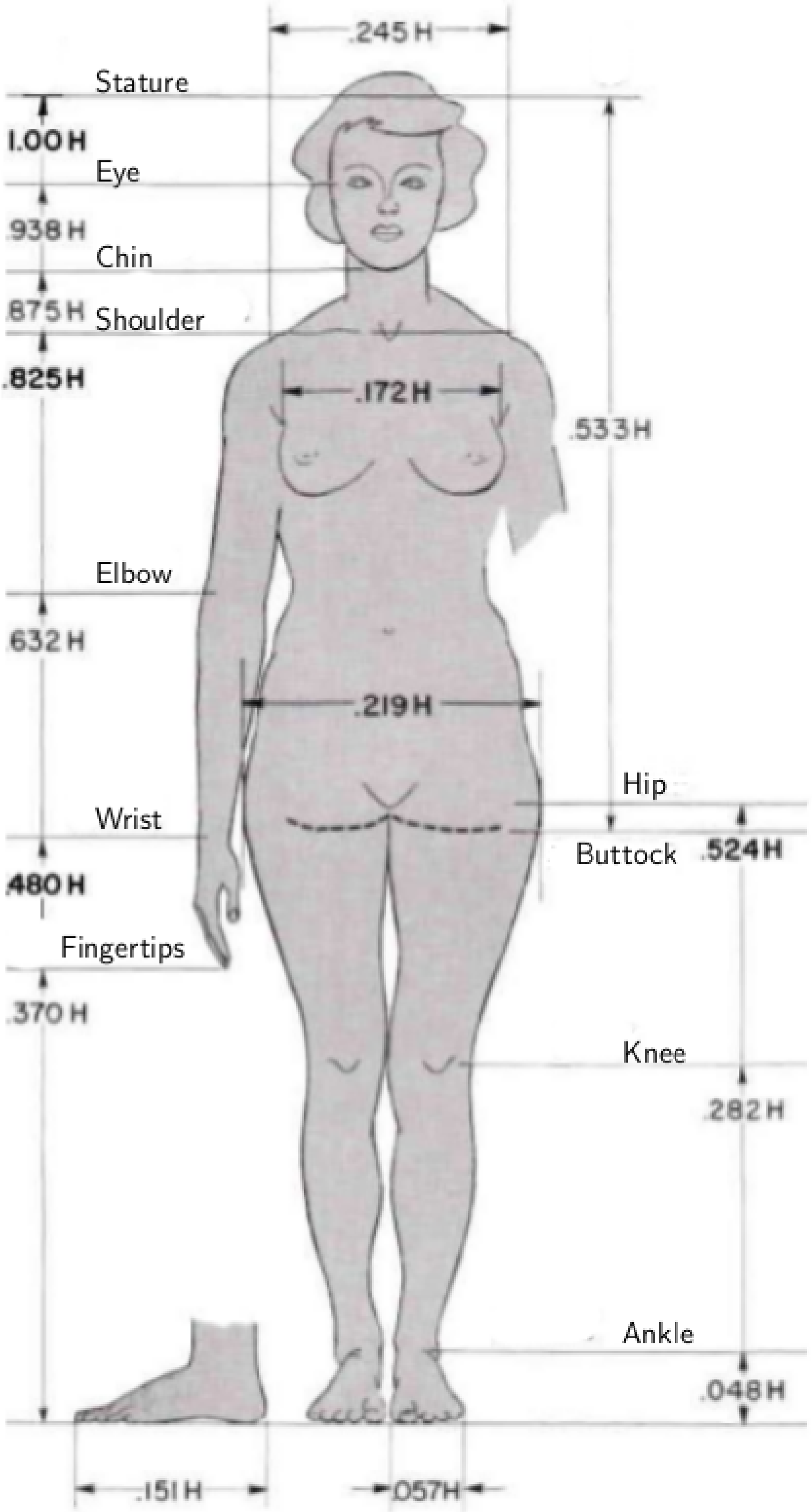

**Figure 5.2**: Female body segment lengths relative to user height. These values are used to calculate the vector lengths for the simplified skeleton [13].

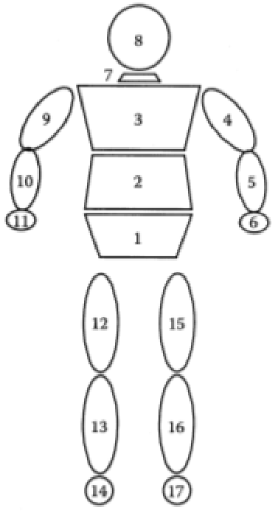

**Figure 5.3**: Reference for the average mass of body segments in **Table 5.9** [14].

**Table 5.1**: Sample row of global orientation correlations to specific location.

| Bone | AHRS 1 | AHRS 2 | AHRS 3 | AHRS 4 |
|---|---|---|---|---|
| Bone 1 | 0.845 | 0.548 | 0.502 | 0.326 |

**Table 5.2**: Sample probability distribution for all AHRS nodes to one location.

| Bone | AHRS 1 | AHRS 2 | AHRS 3 | AHRS 4 |
|---|---|---|---|---|
| Bone 1 | 0.380 | 0.247 | 0.226 | 0.147 |

**Table 5.3**: Sample probability map for four bones and four AHRS nodes.

| Bone | AHRS 1 | AHRS 2 | AHRS 3 | AHRS 4 |
|---|---|---|---|---|
| Bone 1 | 0.380 | 0.247 | 0.226 | 0.147 |
| Bone 2 | 0.068 | 0.191 | 0.506 | 0.235 |
| Bone 3 | 0.102 | 0.623 | 0.102 | 0.173 |
| Bone 4 | 0.068 | 0.204 | 0.155 | 0.875 |

**Table 5.4**: Sample probability map with the most probable highlighted.

| Bone | AHRS 1 | AHRS 2 | AHRS 3 | AHRS 4 |
|---|---|---|---|---|
| Bone 1 | 0.380 | 0.247 | 0.226 | 0.147 |
| Bone 2 | 0.068 | 0.191 | 0.506 | 0.235 |
| Bone 3 | 0.102 | 0.623 | 0.102 | 0.173 |
| Bone 4 | 0.068 | 0.204 | 0.155 | 0.875 |

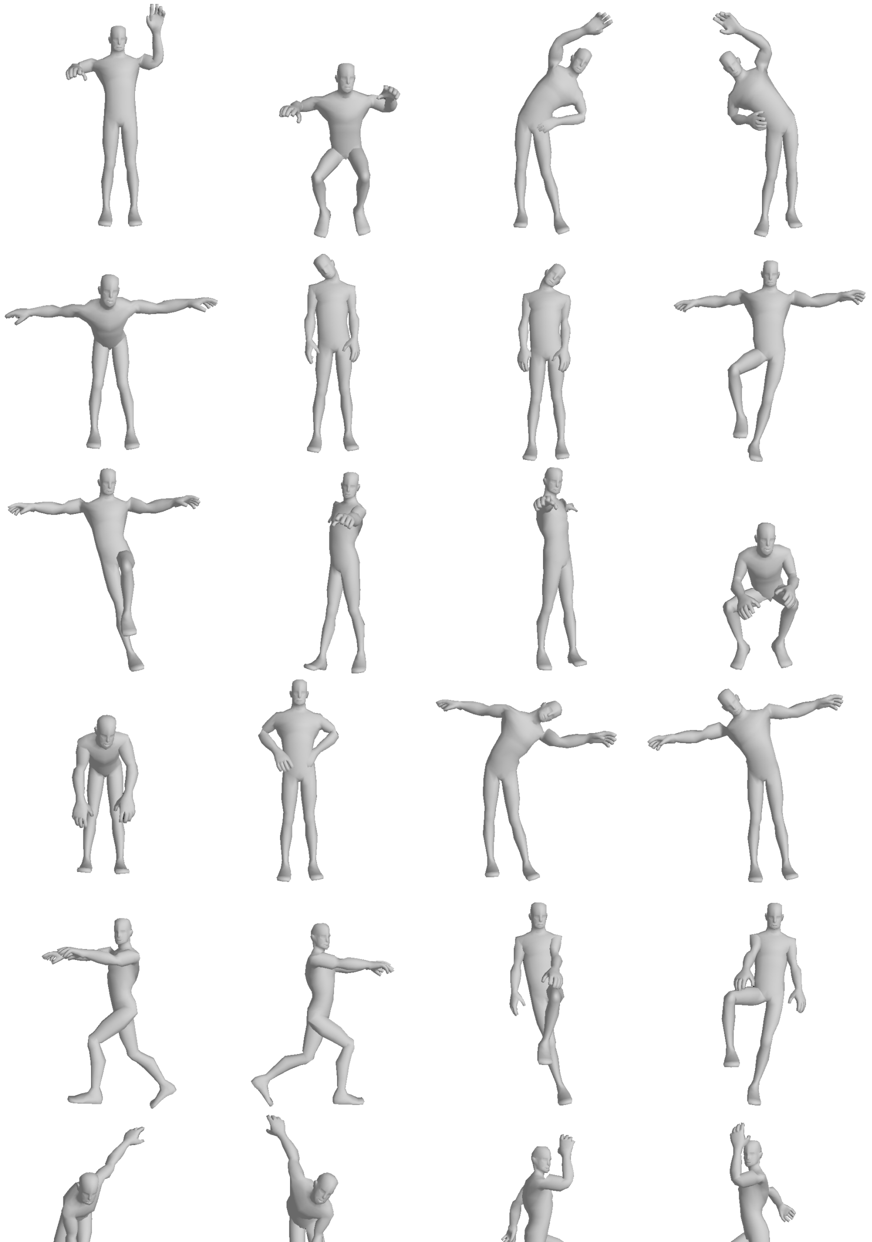

**Figure 5.4**: Twenty-four reference poses to be analyzed. For the remainder of this paper, this figure will be used as a reference to name the poses. As an example, a pose referred to as Pose$_{2,3}$ describes the pose in the second row and third column of this grid (the pose with the head tilted to the right).

| | Pose 1, 1 | Pose 1, 2 | Pose 1, 3 | Pose 1, 4 | Pose 2, 1 | Pose 2, 2 | Pose 2, 3 | Pose 2, 4 | Pose 3, 1 | Pose 3, 2 | Pose 3, 3 | Pose 3, 4 | Pose 4, 1 | Pose 4, 2 | Pose 4, 3 | Pose 4, 4 | Pose 5, 1 | Pose 5, 2 | Pose 5, 3 | Pose 5, 4 | Pose 6, 1 | Pose 6, 2 | Pose 6, 3 | Pose 6, 4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HEAD | | | | | | ○ | ○ | | | | | | | | ○ | ○ | | | | | | | ○ | ○ |
| CHEST | | | ○ | ○ | ○ | | | | | ○ | ○ | ○ | ○ | | ○ | ○ | | | | | ○ | ○ | | |
| STOMACH | | | | | | | | | | ○ | ○ | | ○ | | | | | | | | | | | |
| ARM_R_UPPER | ○ | ○ | ○ | | ○ | | | | | | | | | ○ | | | | | | ○ | | ○ | ○ | ○ |
| ARM_R_LOWER | ○ | | ○ | | ○ | | | | | | | | | ○ | | | | | | ○ | | ○ | ○ | ○ |
| ARM_L_UPPER | ○ | ○ | | ○ | ○ | | | | | | | | | ○ | | | | | ○ | | ○ | | ○ | ○ |
| ARM_L_LOWER | ○ | | | ○ | ○ | | | | | | | | | ○ | | | | | ○ | | ○ | | ○ | ○ |
| LEG_R_UPPER | | ○ | | | | | | ○ | | | | ○ | | | | | ○ | ○ | | ○ | | | ○ | ○ |
| LEG_R_LOWER | | | | | | | | ○ | | | | ○ | | | | | ○ | ○ | | ○ | | | ○ | ○ |
| LEG_L_UPPER | | ○ | | | | | | | ○ | | | ○ | | | | | ○ | ○ | ○ | | | | ○ | ○ |
| LEG_L_LOWER | | | | | | | | | ○ | | | ○ | | | | | ○ | ○ | ○ | | | | ○ | ○ |

**Figure 5.5**: Characterization of all reference poses. A circle represents a body segment that is targeted by the particular pose.

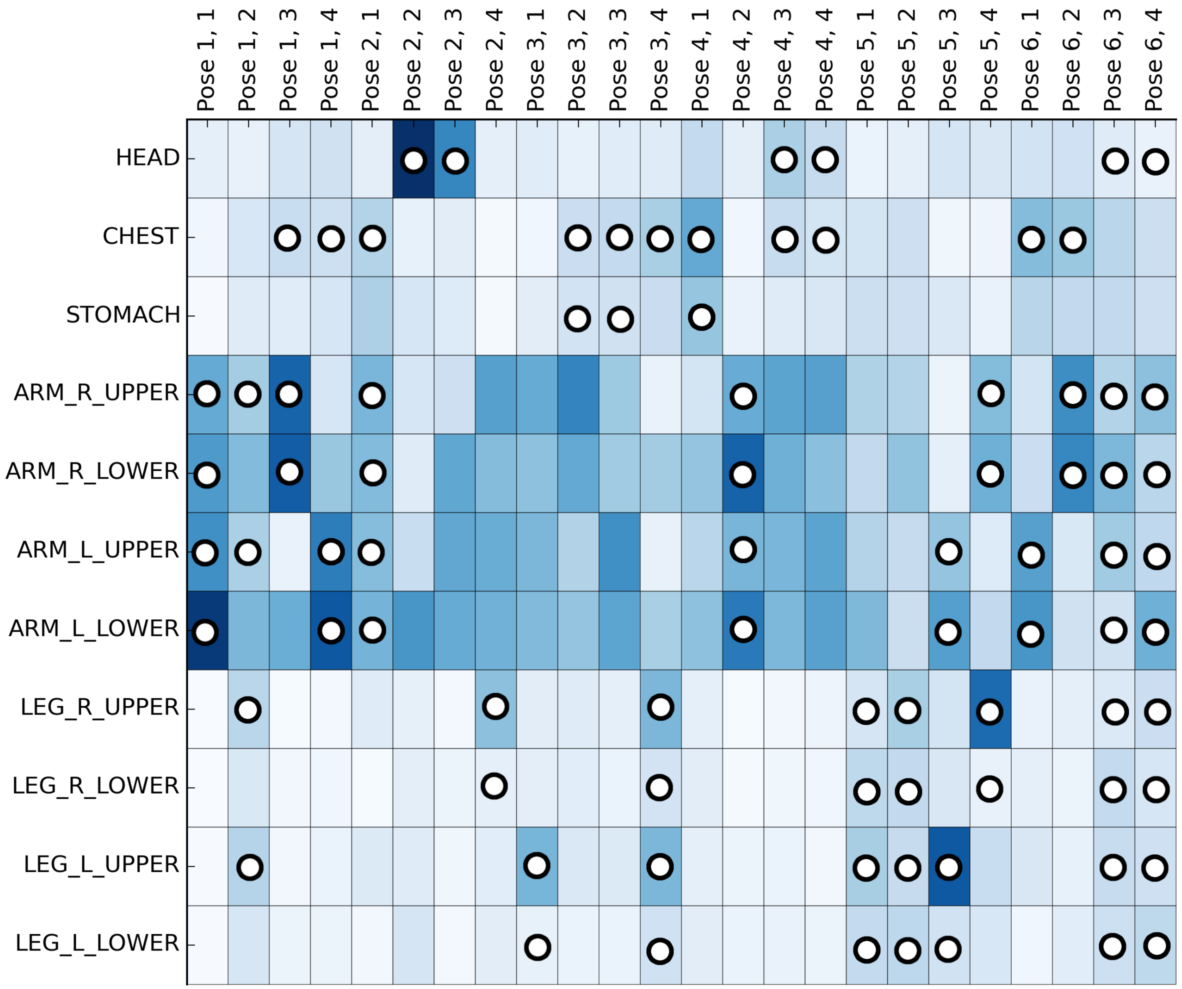

**Figure 5.6**: Pose characterization with values of the disparity between poses.

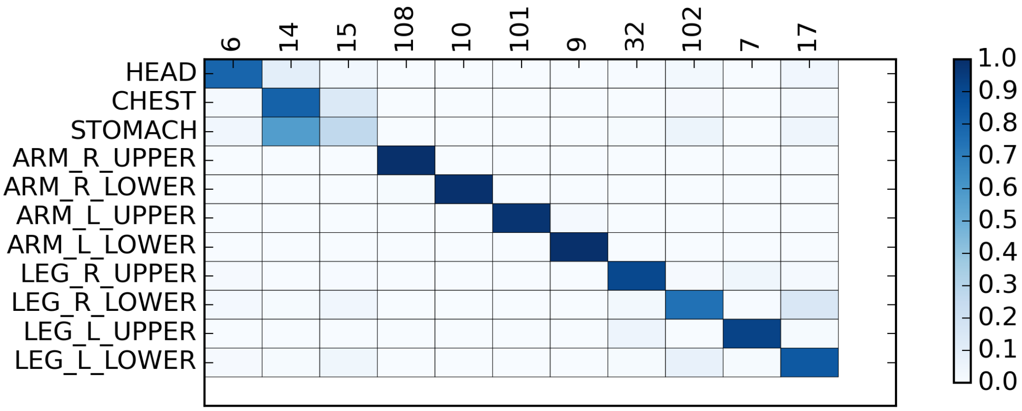

**Figure 5.7**: Sample Probability Map. The column order representing each attached node is sorted so that the correct locations fall along the diagonal.

**Table 5.5**: Probability distributions for the same bone across multiple poses.

| **Pose** | **AHRS 1** | **AHRS 2** | **AHRS 3** | **AHRS 4** |
|---|---|---|---|---|
| Pose 1, Bone 1 | 0.139 | 0.494 | 0.214 | 0.153 |
| Pose 2, Bone 1 | 0.089 | 0.567 | 0.277 | 0.067 |
| Pose 3, Bone 1 | 0.201 | 0.551 | 0.073 | 0.175 |
| Pose 4, Bone 1 | 0.239 | 0.283 | 0.225 | 0.253 |

**Table 5.6**: Poses 1 and 2 from **Table 5.5** multiplied together and renormalized.

| **Pose** | **AHRS 1** | **AHRS 2** | **AHRS 3** | **AHRS 4** |
|---|---|---|---|---|
| Pose 1 × 2 | 0.012 | 0.280 | 0.059 | 0.010 |
| Normalized | 0.034 | 0.774 | 0.164 | 0.028 |

**Table 5.7**: Poses 1, 2, and 3 from **Table 5.5** multiplied together and renormalized.

| **Pose** | **AHRS 1** | **AHRS 2** | **AHRS 3** | **AHRS 4** |
|---|---|---|---|---|
| Pose 1 × 2 × 3 | 0.007 | 0.426 | 0.012 | 0.005 |
| Normalized | 0.015 | 0.947 | 0.027 | 0.011 |

**Table 5.8**: Poses 1, 2, 3, and 4 from **Table 5.5** multiplied together and renormalized.

| **Pose** | **AHRS 1** | **AHRS 2** | **AHRS 3** | **AHRS 4** |
|---|---|---|---|---|
| Pose 1 × 2 × 3 × 4 | 0.004 | 0.268 | 0.006 | 0.003 |
| Normalized | 0.013 | 0.956 | 0.021 | 0.010 |

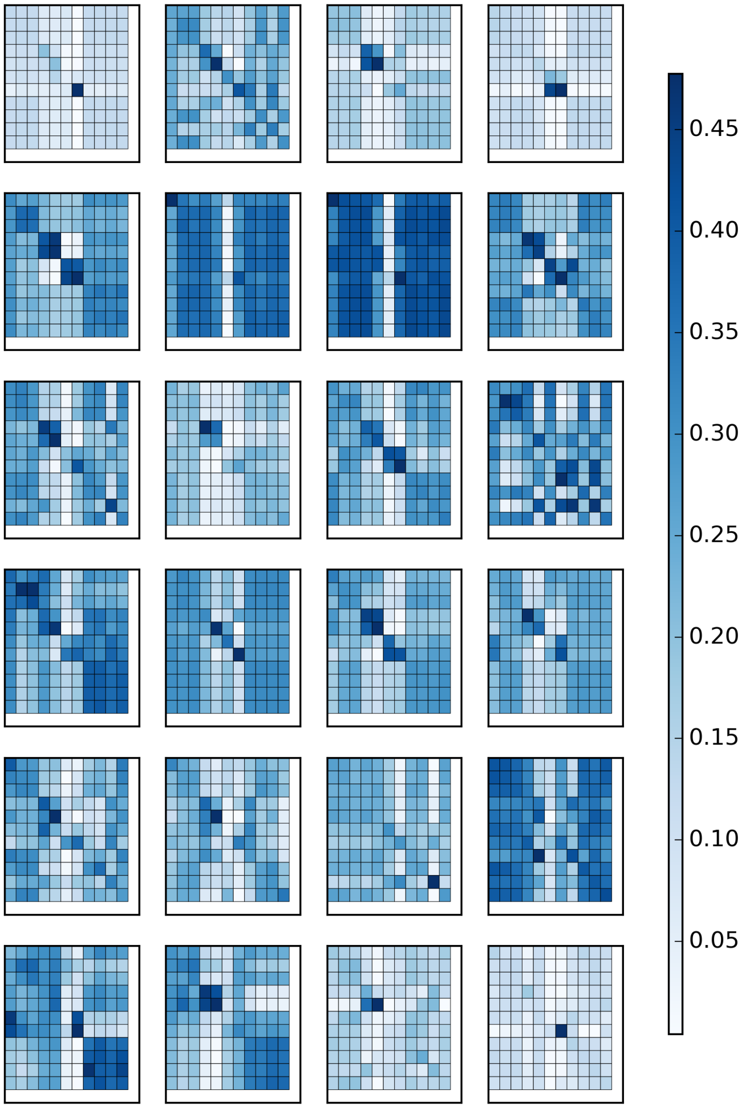

**Figure 5.8**: Individual Probability Maps for the worst-performing subject. The rows and columns of the Probability Maps are ordered so that the correct locations fall along the diagonal. Color has been scaled to the maximum and minimum of the Probability Maps to make differences easier to distinguish.

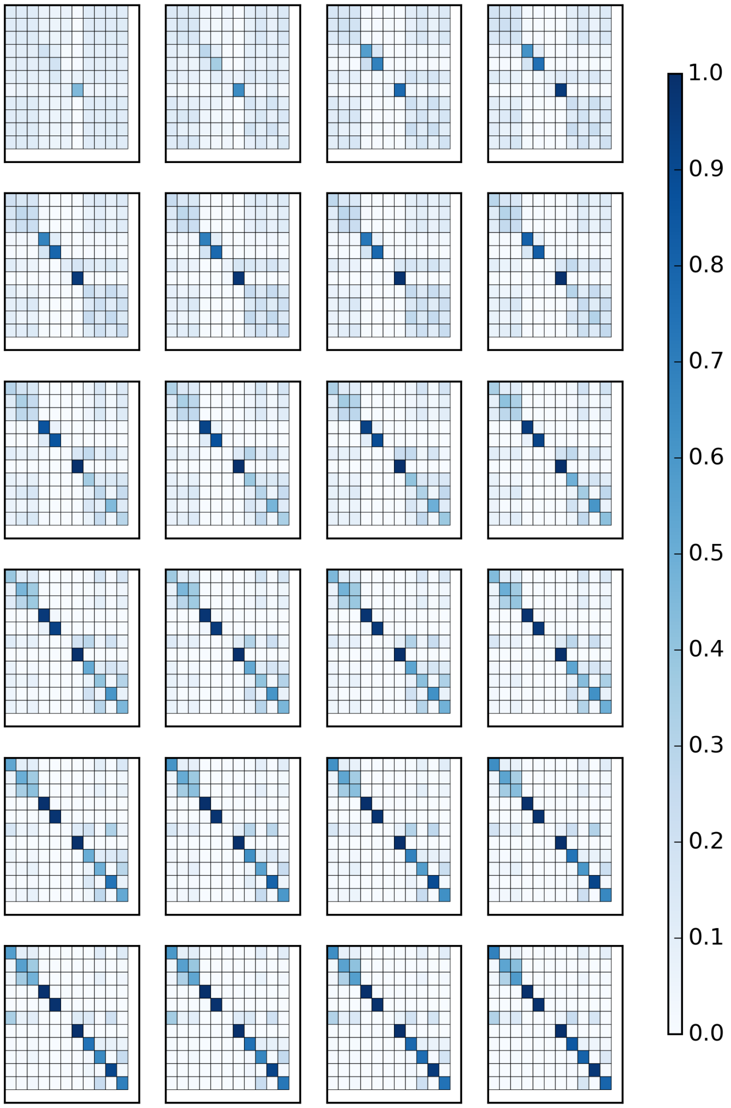

**Figure 5.9**: Progression Probability Map for the worst-performing subject. The rows and columns of the Probability Maps are sorted so that the true location of each AHRS module will fall along the diagonal. Compare with the individual Probability Maps for the same subject in **Figure 5.8**.

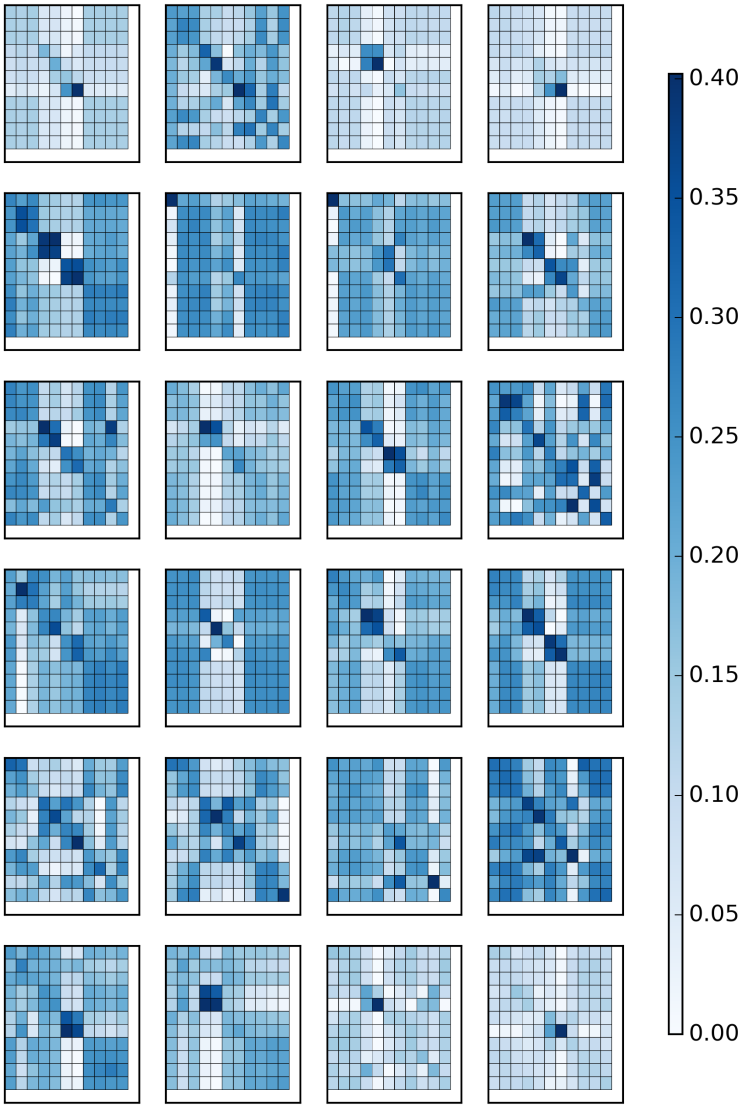

**Figure 5.10**: Individual Probability Maps for the best-performing subject. The rows and columns of the Probability Maps are ordered so that the correct locations fall along the diagonal. Color has been scaled to the maximum and minimum of the Probability Maps to make differences easier to distinguish.

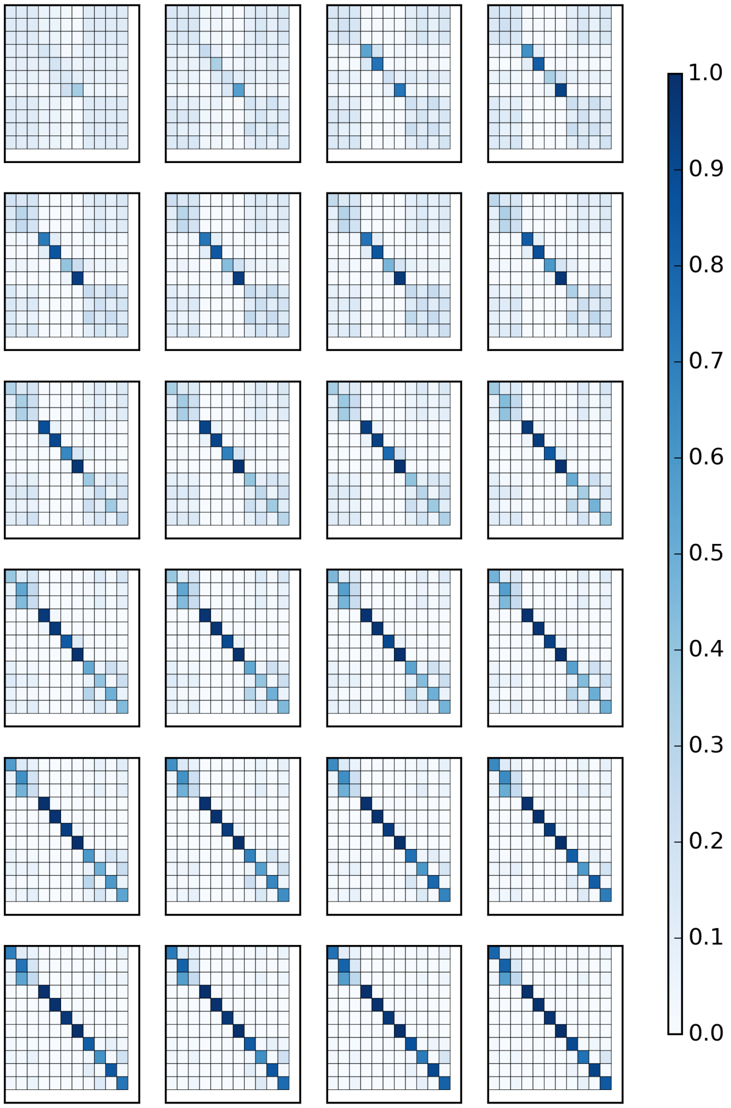

**Figure 5.11**: Progression Probability Map for the best-performing subject. The rows and columns of the Probability Maps are sorted so that the true location of each AHRS module will fall along the diagonal. Compare with the individual Probability Maps for the same subject in **Figure 5.10**.

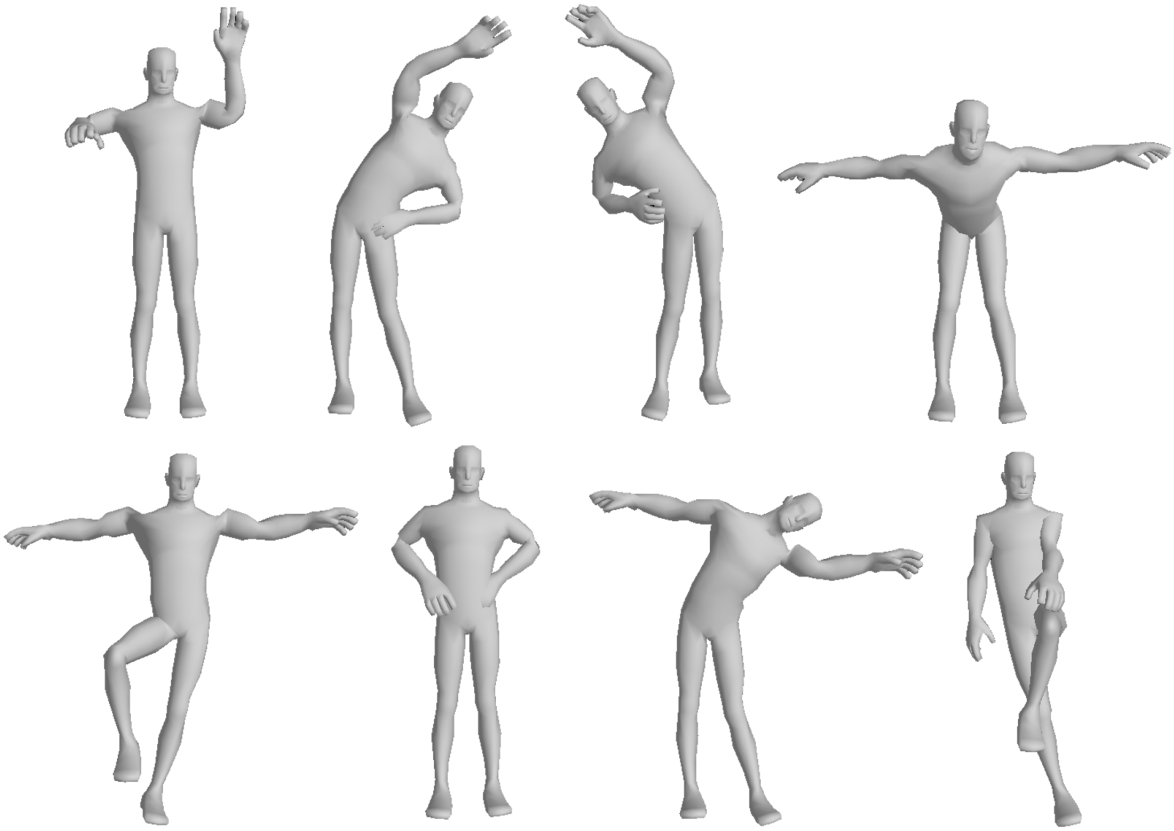

**Figure 5.12**: A set of eight poses that successfully localize all AHRS nodes across all subjects.

**Table 5.9**: Average mass (kg) values for male and female body segments [14]. Calculation of the center of mass requires the percent mass of each segment which can be found by dividing each segment here by the total mass. For body segment locations, see **Figure 5.3**.

| | Body | | Male | | | Female | |
|---|---|---|---|---|---|---|---|
| Name | Segment | 5th% | 50th% | 95th% | 5th% | 50th% | 95th% |
| Waist | 1 | 8.24 | 10.00 | 11.99 | 8.27 | 10.00 | 12.11 |
| Stomach | 2 | 9.01 | 10.95 | 13.13 | 5.45 | 6.59 | 7.98 |
| Chest | 3 | 15.30 | 18.58 | 22.28 | 7.69 | 9.30 | 11.25 |
| Upper arm | 4, 9 | 1.84 | 2.23 | 2.67 | 1.41 | 1.71 | 2.07 |
| Lower arm | 5, 10 | 1.14 | 1.39 | 1.66 | 0.84 | 1.02 | 1.24 |
| Hand | 6, 11 | 0.43 | 0.52 | 0.63 | 0.34 | 0.42 | 0.50 |
| Neck | 7 | 1.48 | 1.80 | 2.16 | 1.20 | 1.45 | 1.76 |
| Head | 8 | 4.07 | 4.95 | 5.93 | 3.31 | 4.01 | 4.85 |
| Upper leg | 12, 15 | 6.96 | 8.45 | 10.13 | 6.22 | 7.53 | 9.11 |
| Lower leg | 13, 16 | 2.84 | 3.45 | 4.14 | 2.24 | 2.71 | 3.28 |
| Foot | 14, 17 | 0.85 | 1.03 | 1.23 | 0.71 | 0.86 | 1.04 |
| Total | | 66.22 | 80.42 | 96.41 | 49.44 | 59.85 | 72.43 |

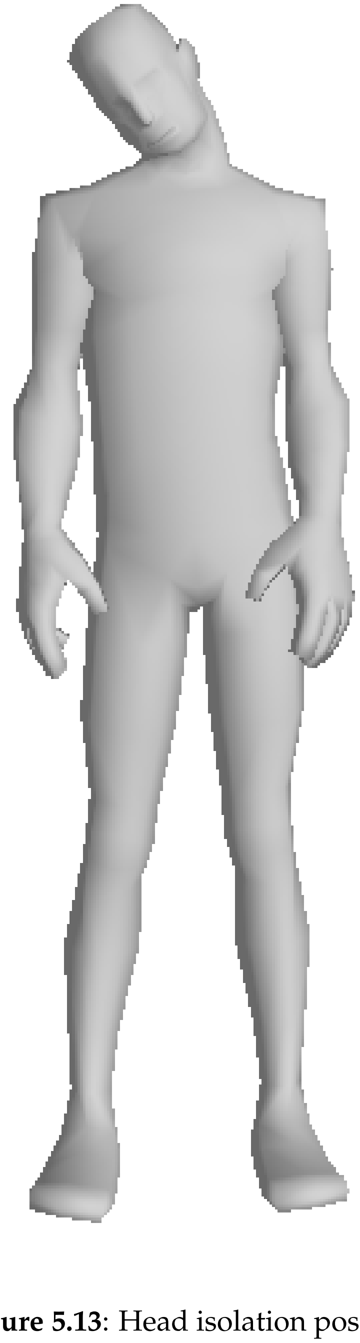

**Figure 5.13**: Head isolation pose.

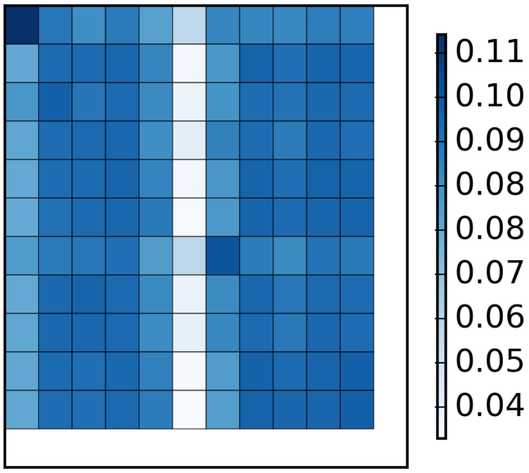

**Figure 5.14**: Even though the Probability Map appears to show a lot of uncertainty, the first column shows that one bone is isolated and a clear match is found.

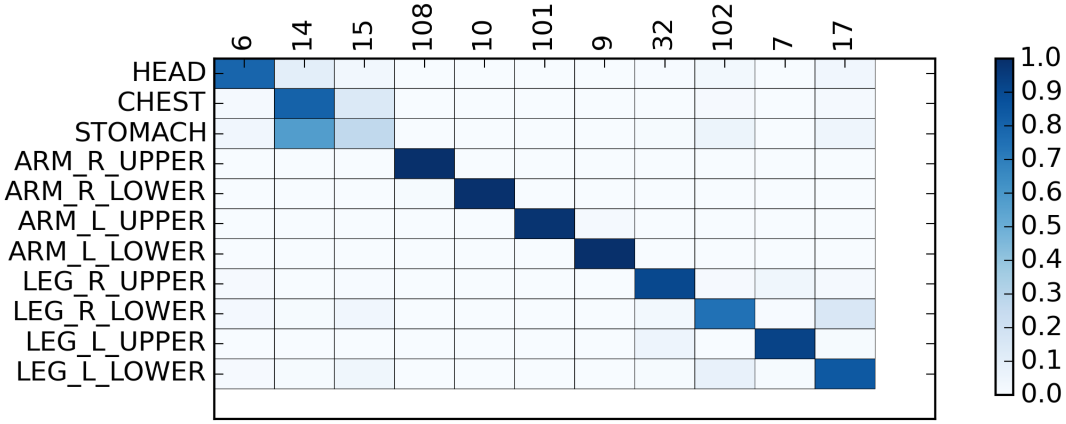

**Figure 5.15**: Final Probability Map for subject 9 after all Probability Maps are multiplied together.

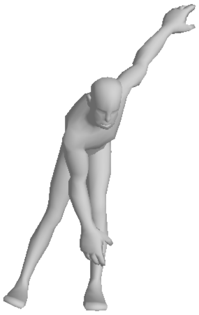

**Figure 5.16**: Pose 21 reference.

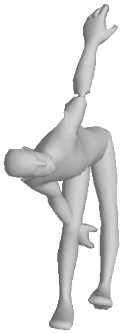

**Figure 5.17**: Pose 21 subject 3.

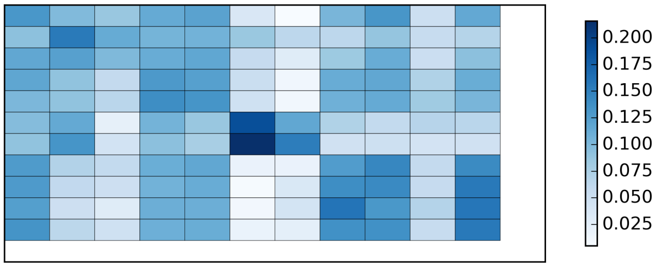

**Figure 5.18**: Comparison between a reference pose and actual output for Pose$_{6,1}$.

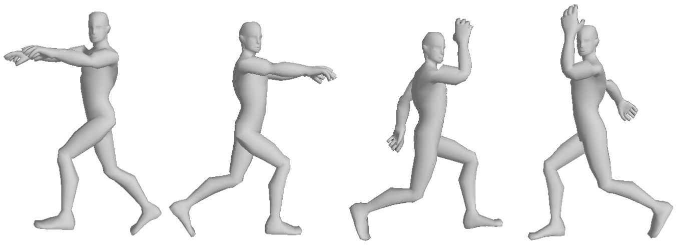

**Figure 5.19**: Four poses that caused confusion for the subjects. These are not instantly clear which leg or arm is back or forward.

**Table 5.10**: Proximal center of mass locations for each major body segment [15]. Each value is given as a percentage of the body segment length. Lengths for each body segment are found from **Figure 5.1** and **Figure 5.2**.

| Segment | Proximal Length |
|---|---:|
| Hand | 50.6% |
| Forearm | 43.0% |
| Upper Arm | 43.6% |
| Forearm and Hand | 68.2% |
| Total Arm | 53.0% |
| Foot | 50.0% |
| Lower Leg (calf) | 43.3% |
| Foot and Lower Leg | 60.6% |
| Upper Leg (thigh) | 43.3% |
| Total Leg | 44.7% |
| Head and Neck | 100.0% |
| Trunk | 50.0% |

**Table 5.11**: Poses with number of targeted locations. This table is arranged to match the ordering of poses that are found in **Figure 5.4**.

| | | | |
|---|---|---|---|
| 4 | 4 | 3 | 3 |
| 5 | 1 | 1 | 2 |
| 2 | 2 | 2 | 5 |
| 2 | 4 | 2 | 2 |
| 4 | 4 | 4 | 4 |
| 3 | 3 | 9 | 9 |

**Table 5.12**: Number of poses that intentionally isolate each bone.

| Location | Appearances |
|---|---|
| Head | 6 |
| Chest | 11 |
| Stomach | 3 |
| Arm R Upper | 9 |
| Arm R Lower | 8 |
| Arm L Upper | 9 |
| Arm L Lower | 8 |
| Leg R Upper | 8 |
| Leg R Lower | 7 |
| Leg L Upper | 8 |
| Leg L Lower | 7 |

**Table 5.13**: Final Localizer output for each attached AHRS for each subject. Values that are equal to the header location mean the correct location was found.

| | Head | Chest | Stomach | Arm R Upper | Arm R Lower | Arm L Upper | Arm L Lower | Leg R Upper | Leg R Lower | Leg L Upper | Leg L Lower |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Subject 1 | Head | Chest | Stomach | Arm R Upper | Arm R Lower | Arm L Upper | Arm L Lower | Leg R Upper | Leg R Lower | Leg L Upper | Leg L Lower |
| Subject 2 | Head | Chest | Stomach | Arm R Upper | Arm R Lower | Arm L Upper | Arm L Lower | Leg R Upper | Leg R Lower | Leg L Upper | Leg L Lower |
| Subject 3 | Head | Chest | Stomach | Arm R Upper | Arm R Lower | Arm L Upper | Arm L Lower | Leg R Upper | Leg R Lower | Leg L Upper | Leg L Lower |
| Subject 4 | Head | Chest | Stomach | Arm R Upper | Arm R Lower | Arm L Upper | Arm L Lower | Leg R Upper | Leg R Lower | Leg L Upper | Leg L Lower |
| Subject 5 | Head | Chest | Stomach | Arm R Upper | Arm R Lower | Arm L Upper | Arm L Lower | Leg R Upper | Leg R Lower | Leg L Upper | Leg L Lower |
| Subject 6 | Head | Chest | Stomach | Arm R Upper | Arm R Lower | Arm L Upper | Arm L Lower | Leg R Upper | Leg R Lower | Leg L Upper | Leg L Lower |
| Subject 7 | Head | Chest | Stomach | Arm R Upper | Arm R Lower | Arm L Upper | Arm L Lower | Leg R Upper | Leg R Lower | Leg L Upper | Leg L Lower |
| Subject 8 | Head | Chest | Stomach | Arm R Upper | Arm R Lower | Arm L Upper | Arm L Lower | Leg R Upper | Leg R Lower | Leg L Upper | Leg L Lower |
| Subject 9 | Head | Chest | Stomach | Arm R Upper | Arm R Lower | Arm L Upper | Arm L Lower | Leg R Upper | Leg R Lower | Leg L Upper | Leg L Lower |
| Subject 10 | Head | Chest | Stomach | Arm R Upper | Arm R Lower | Arm L Upper | Arm L Lower | Leg R Upper | Leg R Lower | Leg L Upper | Leg L Lower |
| Subject 11 | Head | Chest | Stomach | Arm R Upper | Arm R Lower | Arm L Upper | Arm L Lower | Leg R Upper | Leg R Lower | Leg L Upper | Leg L Lower |
| Subject 12 | Head | Chest | Stomach | Arm R Upper | Arm R Lower | Arm L Upper | Arm L Lower | Leg R Upper | Leg R Lower | Leg L Upper | Leg L Lower |

**Table 5.14**: Subject ranking for how ideal the data collected matches the reference pose.

| Subject | Ranking |
|---|---|
| Subject 1 | 1.07e-4 |
| Subject 2 | 3.58e-2 |
| Subject 3 | 7.36e-7 |
| Subject 4 | 9.24e-3 |
| Subject 5 | 3.60e-2 |
| Subject 6 | 1.84e-4 |
| Subject 7 | 7.51e-2 |
| Subject 8 | 6.44e-2 |
| Subject 9 | 8.75e-2 |
| Subject 10 | 2.97e-2 |
| Subject 11 | 8.38e-2 |
| Subject 12 | 2.14e-2 |

# CHAPTER 6

# CONCLUSION

The MEMS sensors needed to make an AHRS were properly calibrated using an inexpensive software-based method. These sensors were then used to create a sufficiently accurate AHRS module that could be used for completely mobile motion capture. In order to make the system more consumer-friendly, a localization method was created.

## 6.1 Discussion

The design and implementation of a low-cost, mobile, and scalable motion capture system successfully resulted in a calibration method, sufficient AHRS accuracy with the ability to improve accuracy with redundant nodes, and a localization framework enabling arbitrary node placement. A software-based calibration technique to transform raw MEMS gyroscope, accelerometer, and magnetometer values into real-world units was successfully implemented and verified. The correct calibration of these sensors allowed the creation of a custom AHRS node for mobile motion capture purposes. The efficacy of these AHRS modules was researched for individual units as well as the accuracy gains that come from putting redundant modules on the same rigid body. After the accuracy of the global orientations provided by the AHRS nodes proved to be sufficient for uses of a mobile motion capture system, a technique to quickly initialize the system was created. An algorithm to allow a user to place nodes in arbitrary locations and move through a set of poses to localize the AHRS units was designed and tested. All of these pieces together created a mobile motion capture system based on AHRS modules that is inexpensive and intuitive enough to be used by consumers.

Chapter 2 describes the techniques to calibrate the MEMS gyroscopes, accelerometers, and magnetometers. For gyroscopes, even long integration times of up to 10-s produced final rotational position values within 6.45 degrees. Shorter integration times of 1-s produced results within 0.80 degrees of the true value. Because the AHRS units use the gyroscopes as transitional values between updates from the accelerometer and magnetometer references, only small integration periods are used which makes these results more than acceptable.

Accelerometers produced correlation values between 83.7% and 86.4%. This makes the calibrated accelerometers useful as a low-cost tilt sensor for balancing applications. For the purposes of AHRS motion capture, the accelerometers are used as a reference value much like a tilt sensor, and these results are sufficient.

The magnetometers needed to be calibrated with both hard- and soft-iron distortions present. After the ellipsoidal Gauss-Newton fitting algorithm was employed, very high accuracy of the magnetic field heading were recorded. Depending on which correlation value is used, accuracy ranged between 89.8% and 98.8% with most values above 97%. The magnetometers are used as a reference much like the accelerometers so this calibration procedure exceeds the accuracy needed for AHRS motion capture.

Chapters 3 and 4 outline a method to build an AHRS from the calibrated MEMS sensors and the effectiveness of redundant units was explored. Single AHRS nodes experienced gaps or connection that caused errors to appear in the final data. This was due to a slow serial connection being used for data capture. However, it was shown that redundant sensors overcame this limitation. Using the specialized averaging technique employed in Chapter 3, the redundant nodes eliminated these errors. Further averaging techniques presented in Chapter 4 produced very accurate global rotational values. An accuracy spread between 85.2% and 99.6% were detected. It should be noted that the lowest correlation values were found in small amounts of time less than 1.5s. By moving to a faster collection framework, these problems can be remedied.

Chapter 5 provides a method to easily initialize a motion capture system based on AHRS units. In order to make the motion capture system intuitive enough that it can be used by consumers without prior training, a localization algorithm, called the *Localizer*, was developed to allow users to arbitrarily place AHRS nodes without manually entering their locations. A bank of reference poses was designed that were used to build *Probability Maps*. These Probability Maps were multiplied together to cause each attached AHRS node to converge to its true location. For each of the twelve subjects, the Localizer successfully placed all attached AHRS modules. Further, a subset of eight poses was found that would still successfully localized all units for all twelve subjects.

The Localizer proved robust enough to successfully localize nodes even when subjects did a poor job of matching the reference poses or when the base pose was poorly performed. This localization algorithm makes an AHRS-based motion capture system likely to perform well for non-technical consumers, and does not require manually inputing the locations of an attached AHRS unit.

The success of this work has shown that AHRS motion capture is a viable reality. The techniques employed in the creation of this system allow it to be low-cost, intuitive and mobile without the need for a dedicated studio or technician.

## 6.2 Future work

While researching the Localizer algorithm, some interesting questions have arisen that will serve as the basis for future work. A set of twenty-four poses was used for reference and although these poses proved to be effective in localizing the AHRS nodes, there exists an *optimal* group of poses that will successfully locate AHRS units across subjects. A subset of only eight poses can be used to localize for an entire body and one such set was identified. It is possible that there are other subsets that prove to be more robust or are easier for users to perform. Evaluation of the poses in such a subset may further suggest different poses that may be even more effective, further reducing the number of poses required.

There was no consideration placed on how difficult each pose was to perform. A study to compare the effectiveness of poses with the difficulty to mimic those poses should be performed. Further, what poses could be used for users that are prevented from using the localization technique due to disabilities or missing limbs?

The Localizer was not evaluated with redundant nodes on bones. Because it has been shown that redundant nodes can enhance the accuracy of AHRS motion capture, the Localizer should be tested with cases where there is an unknown number of connected AHRS units. The accuracy of the Localizer could potentially decrease as larger and larger networks of redundant nodes are attached to the user.

There are limits for how many nodes can be successfully connected. A study for where those limits exist and what steps can be taken to increase them should be performed.

Currently, AHRS motion capture has been studied for the human body. One advantage to AHRS motion capture is that it can be scaled to new forms. By adding an object as additional limb such as a golf club, a new forward kinematic skeleton can be employed and track more than the human body. AHRS motion capture has potential to be employed outside of humans such as robots or animals.

## 6.3 Conclusion

The investigation into AHRS motion capture was a success. By calibrating MEMS gyroscopes, accelerometers, and magnetometers and using these calibrated sensors to build wireless AHRS nodes, a motion capture system that solves the limitations of optically-based systems was created. The low-cost nature of the MEMS sensors employed in making the AHRS modules allows the system to be used by consumers or in situations were motion capture was not possible.

The Localizer allows the system to be deployed with arbitrary node placement, reducing the need for training. The mobile nature of wireless nodes and ease of setup opens new segments to the benefits of motion capture. Motion capture that needs to be performed outside of a studio such as downhill skiing now has a viable solution. AHRS motion capture changes the landscape of customers and applications for motion capture.

# APPENDIX A

# EULER ANGLE REPRESENTATIONS OF AHRS ACCURACY

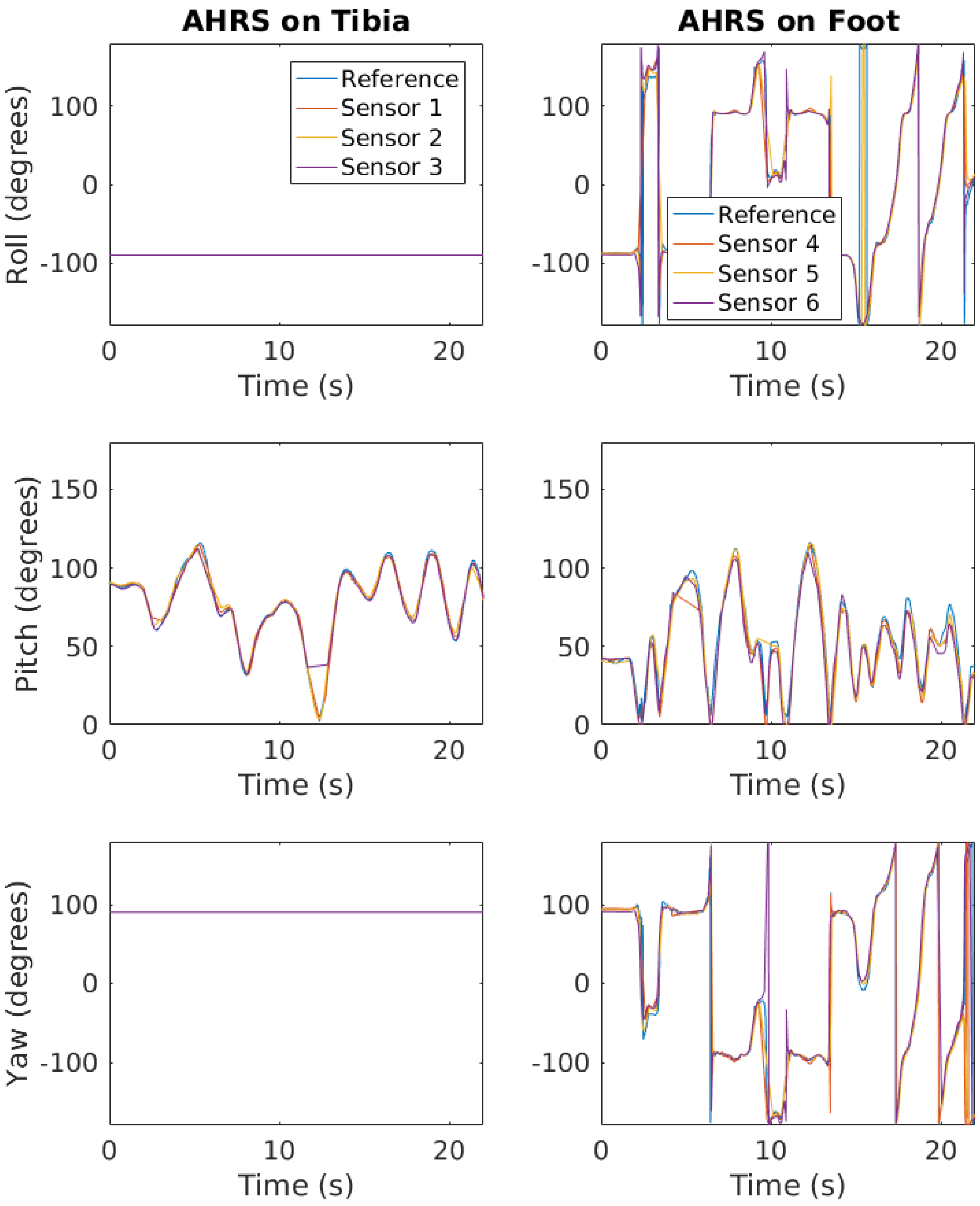

**Figure A.1**: Euler angles of all AHRS units for both the tibia and foot section of the mechanical leg.

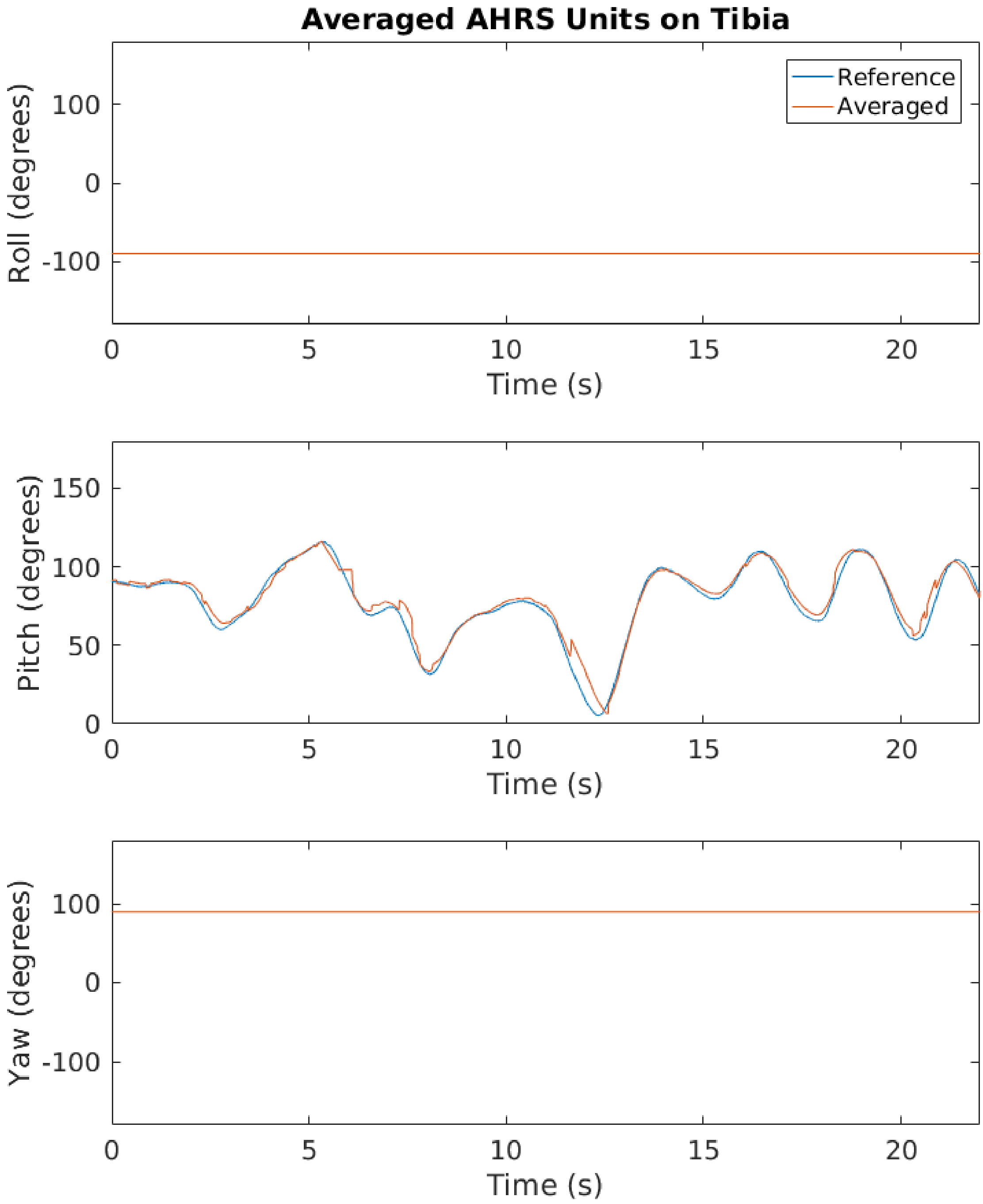

**Figure A.2**: Averaged euler angles for the tibia section of the mechanical leg. The roll and yaw anglse are constant because they were used as a basis for calibration.

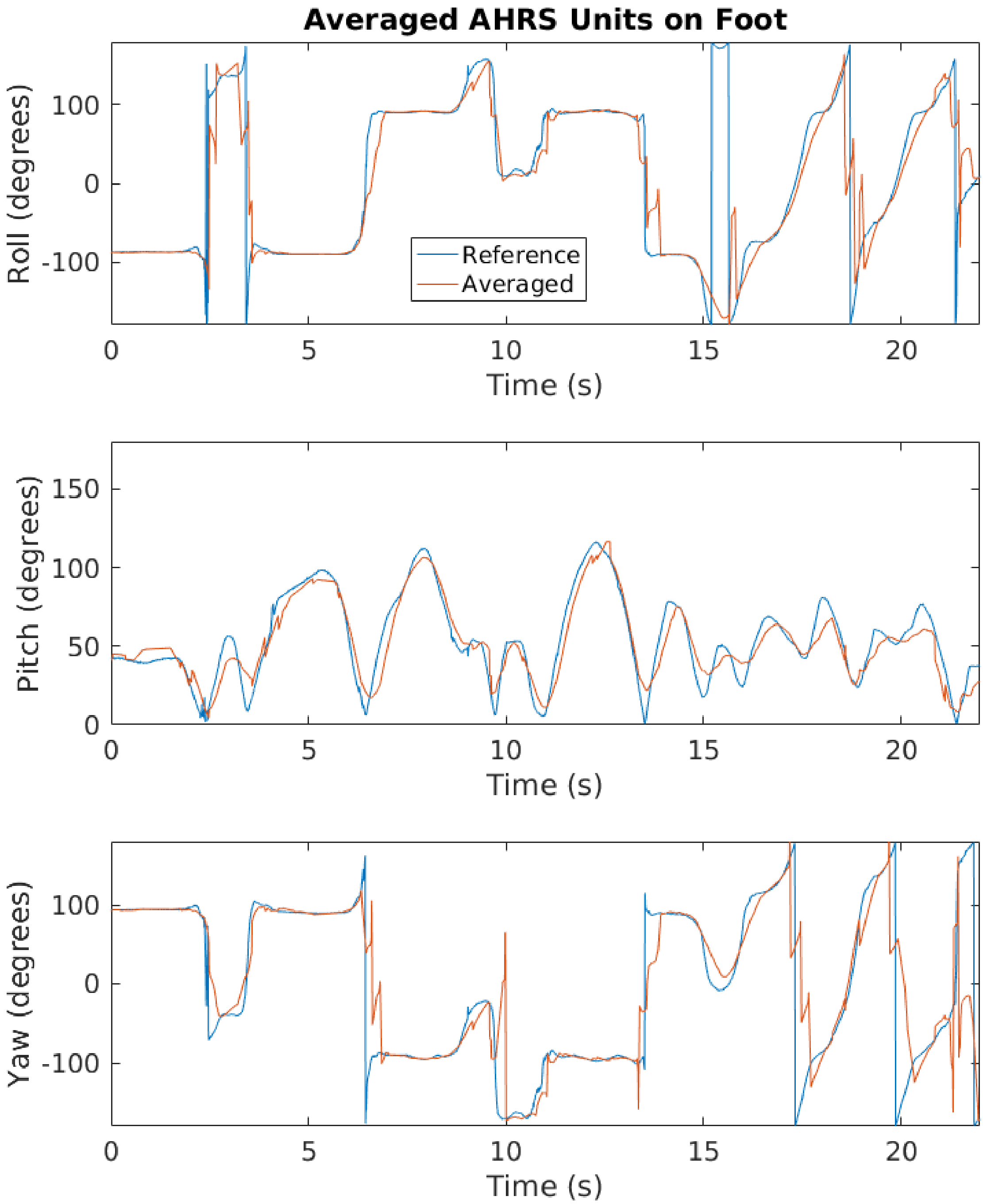

**Figure A.3**: Averaged euler agnles for the foot section of the mechanical leg.

# APPENDIX B

# INDIVIDUAL PROBABILITY MAPS

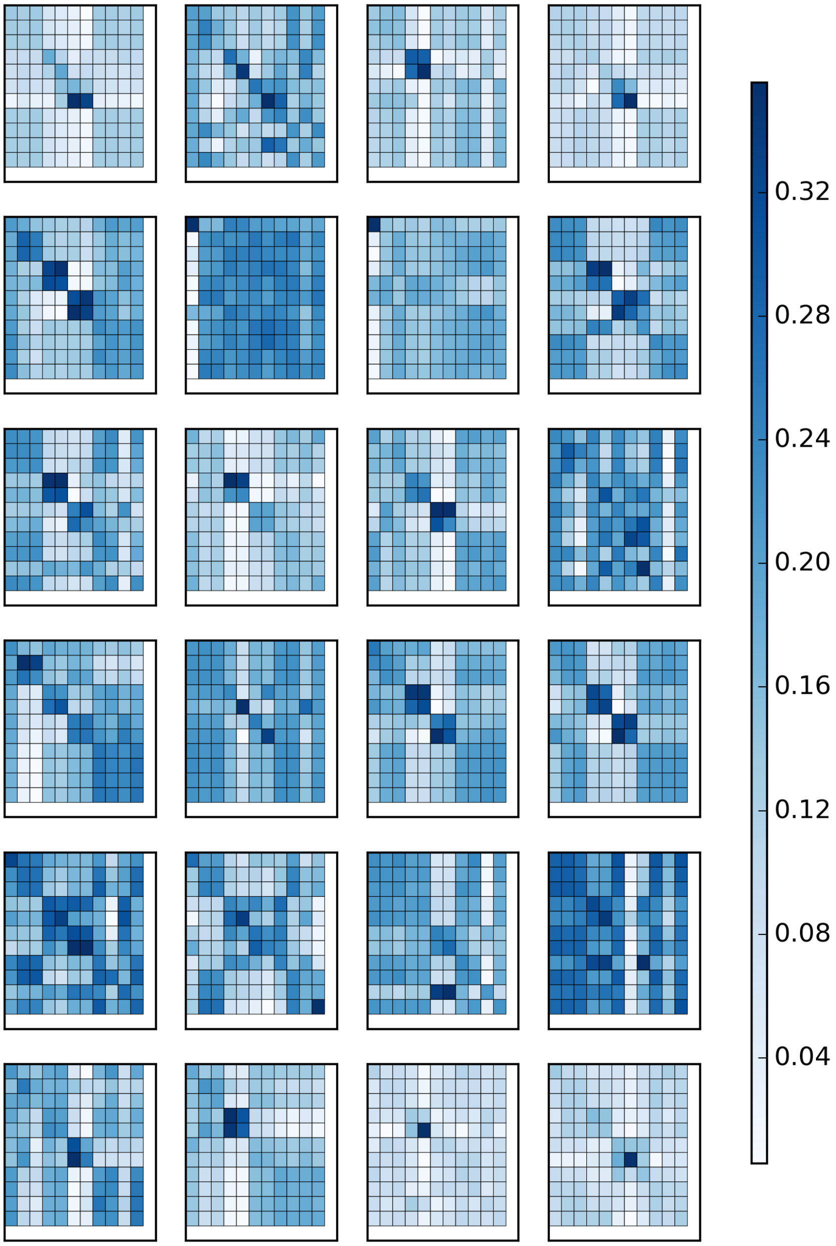

**Figure B.1**: Individual Probability Maps for subject 1.

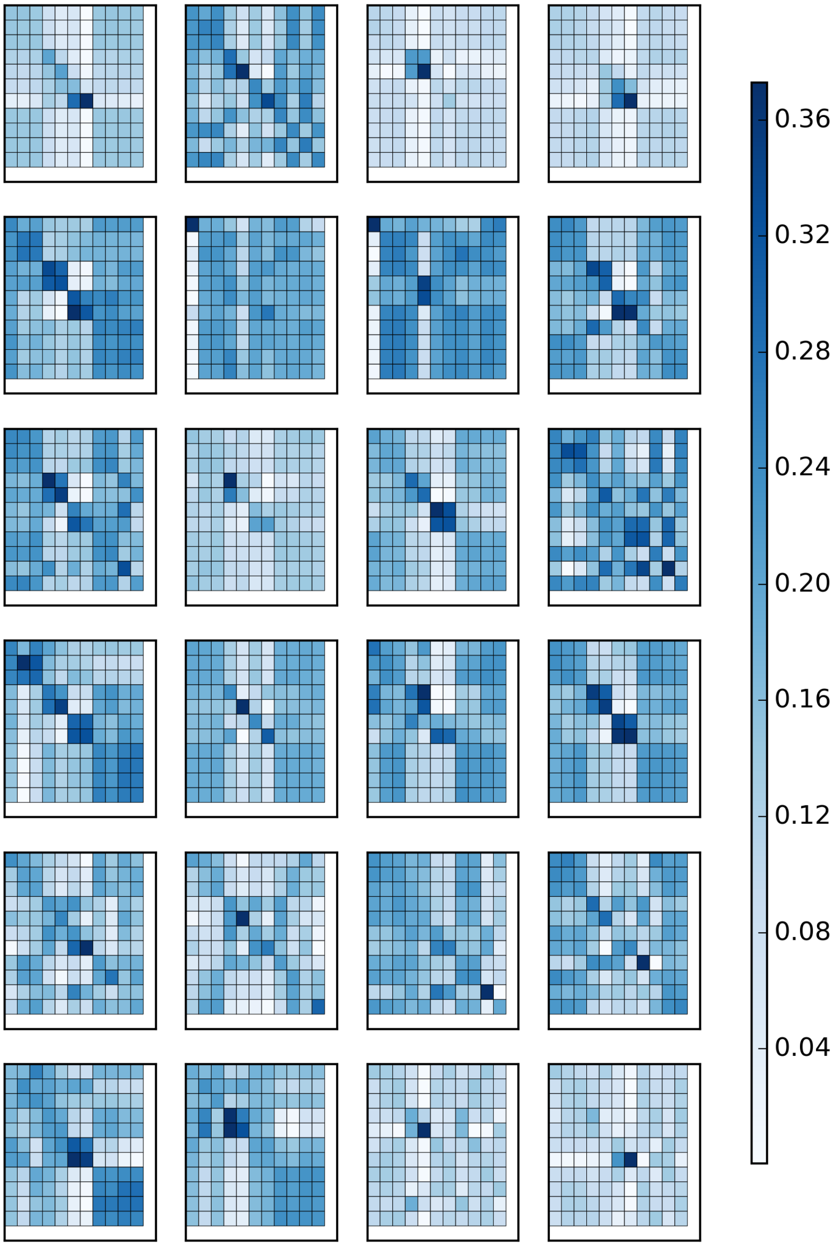

**Figure B.2**: Individual Probability Maps for subject 2.

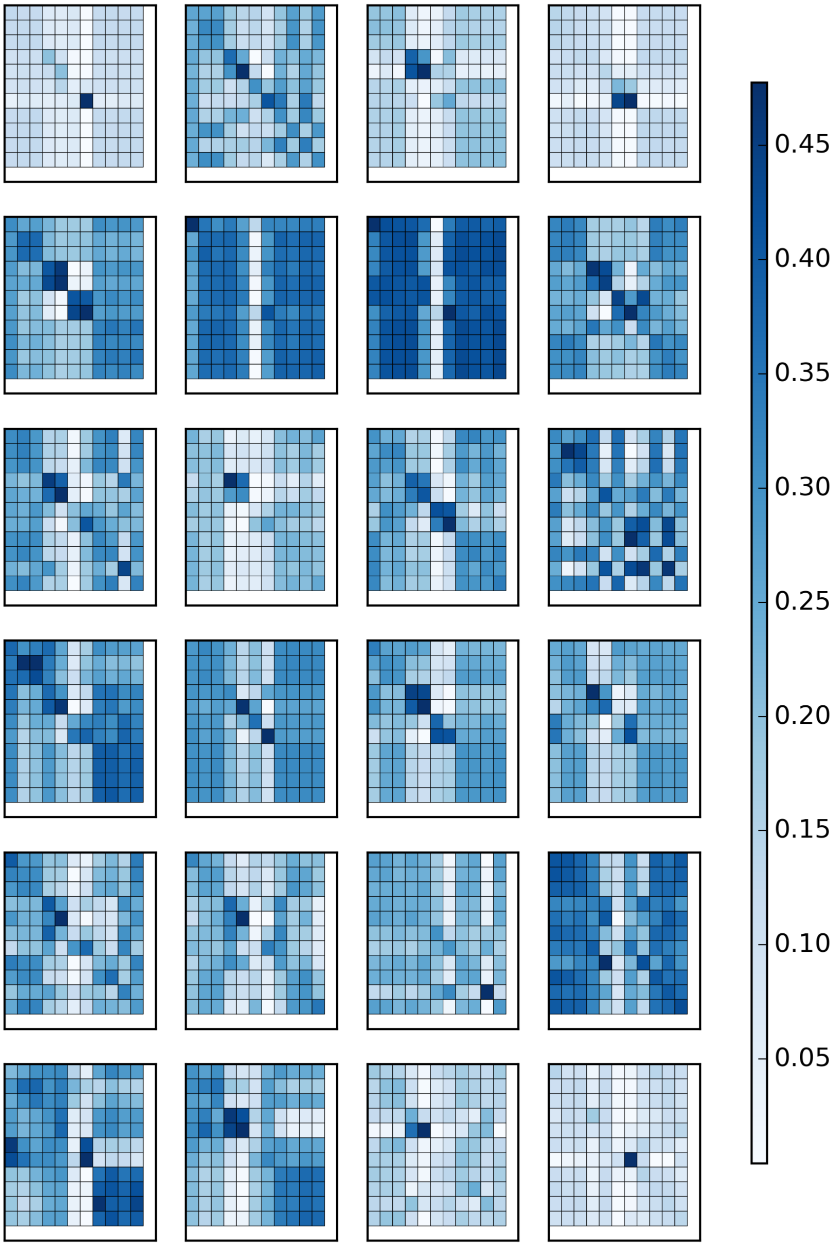

**Figure B.3**: Individual Probability Maps for subject 3.

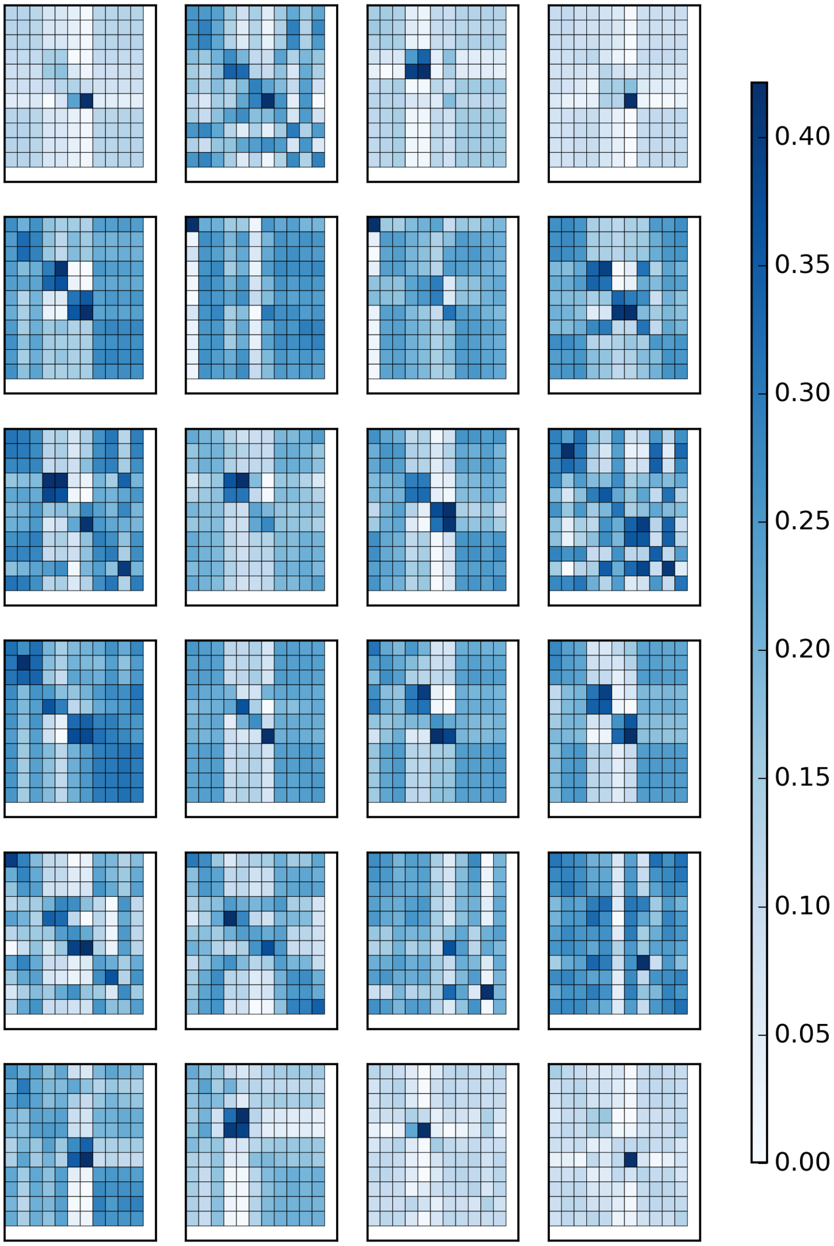

**Figure B.4**: Individual Probability Maps for subject 4.

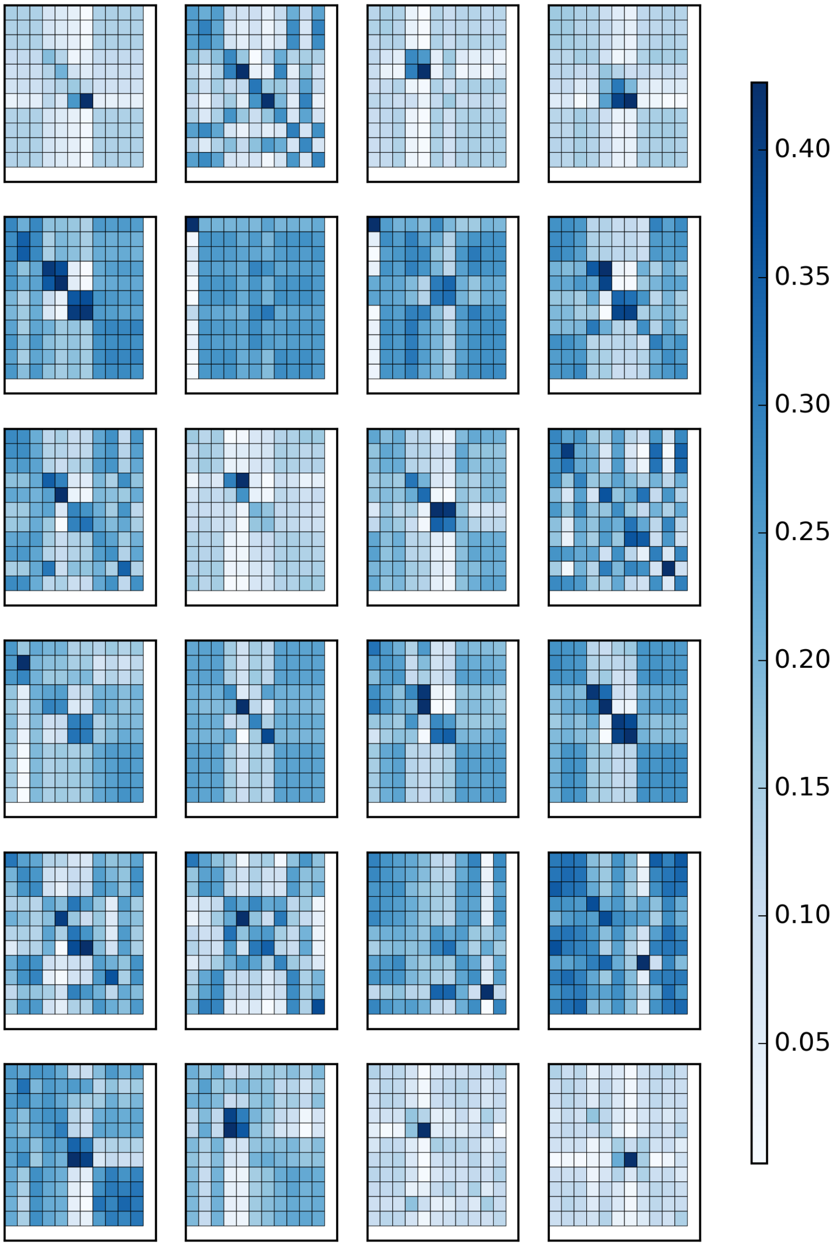

**Figure B.5**: Individual Probability Maps for subject 5.

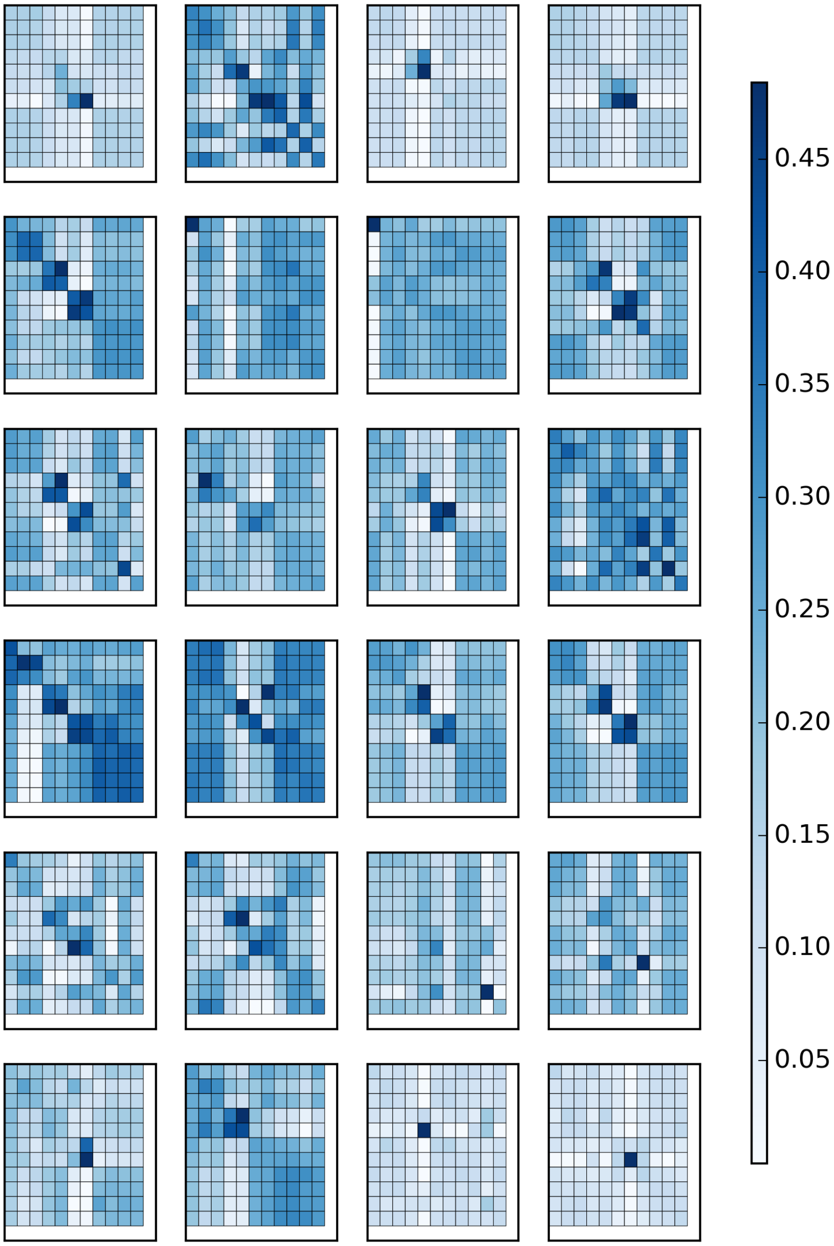

**Figure B.6**: Individual Probability Maps for subject 6.

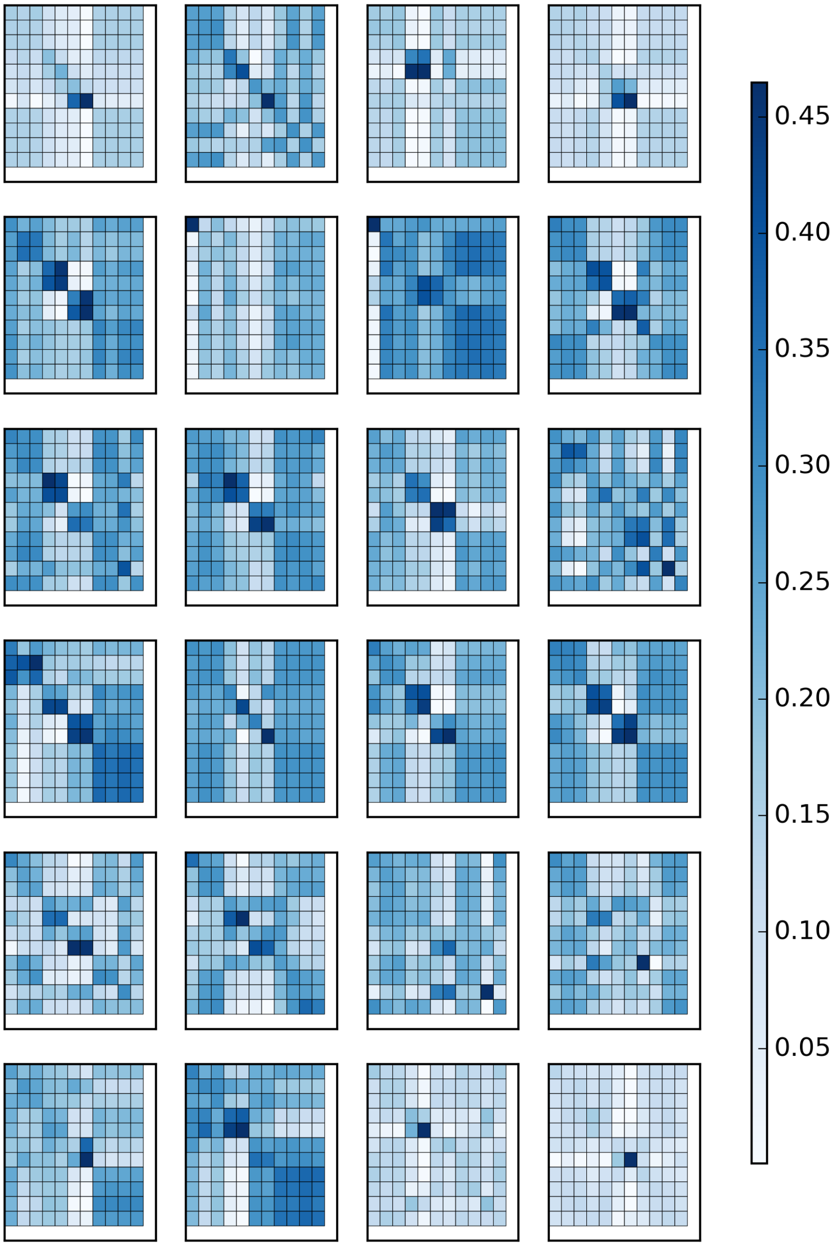

**Figure B.7**: Individual Probability Maps for subject 7.

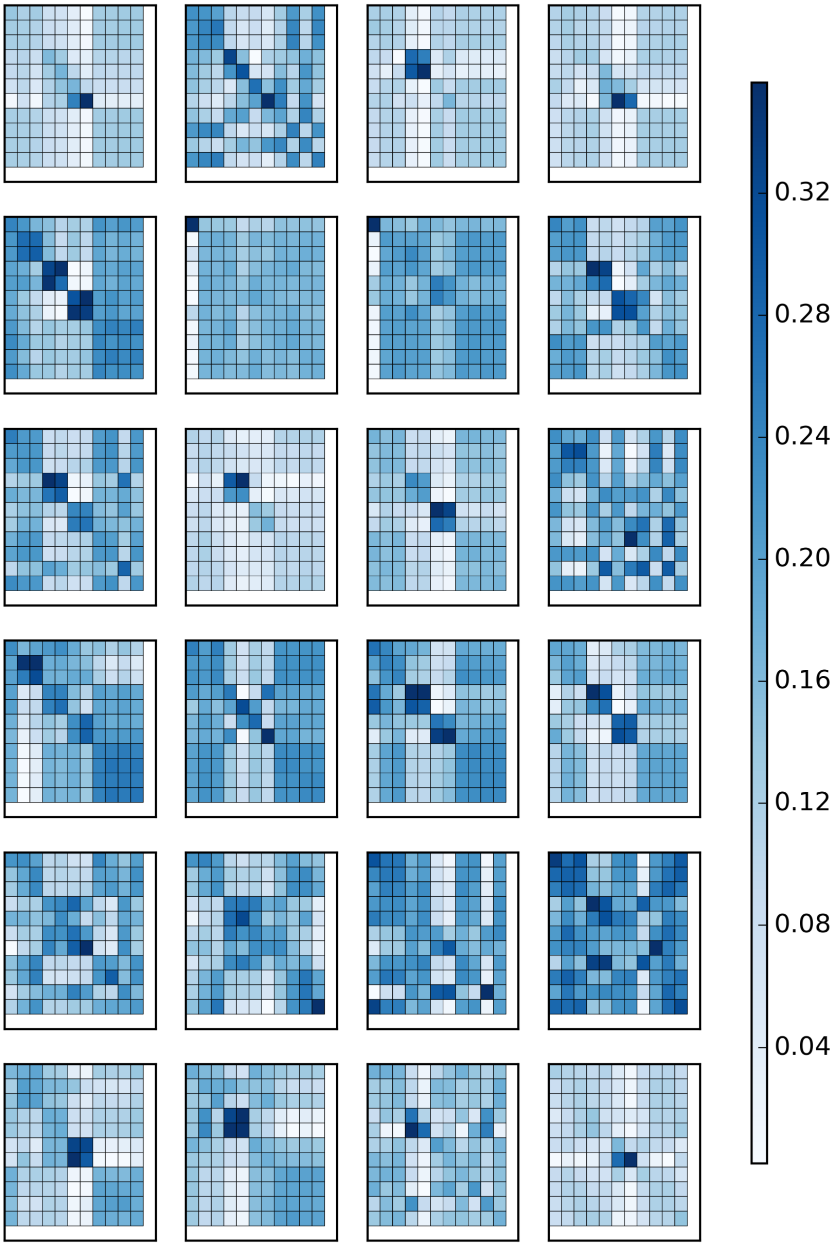

**Figure B.8**: Individual Probability Maps for subject 8.

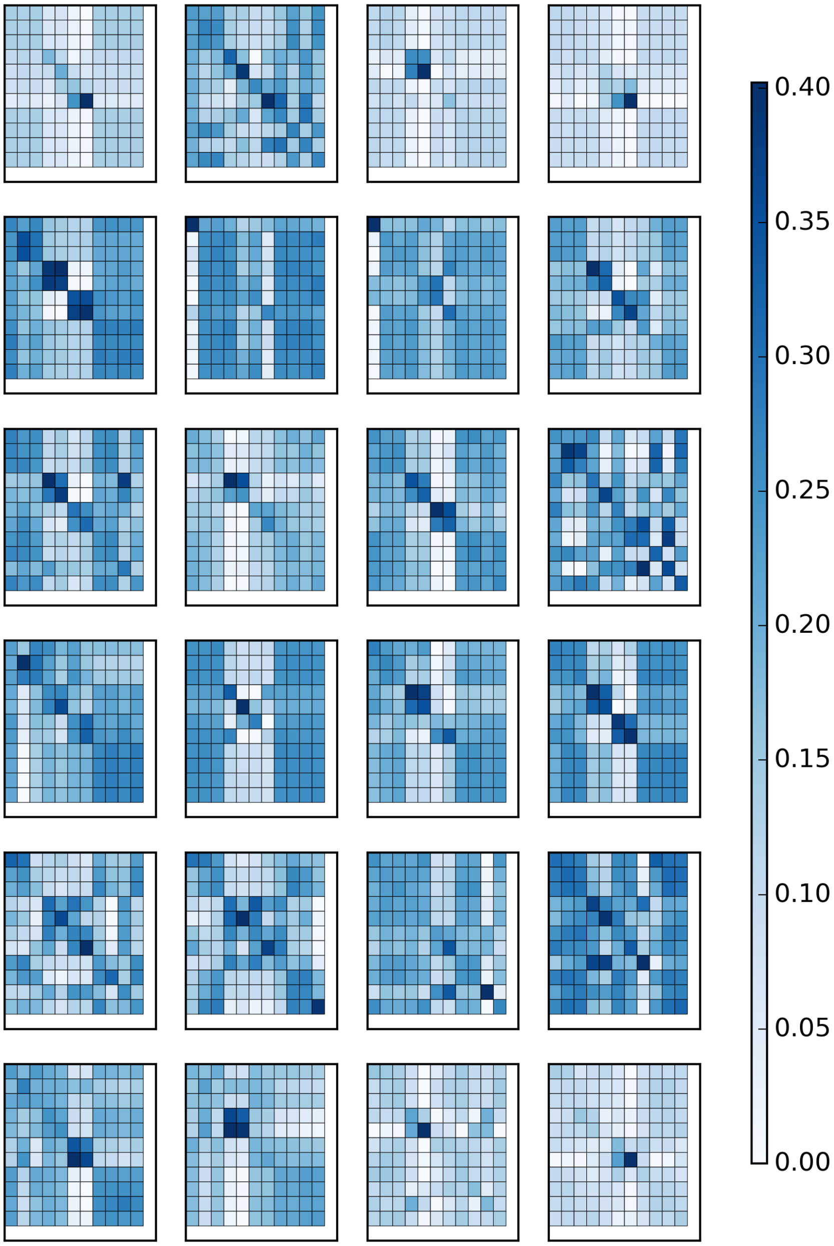

**Figure B.9**: Individual Probability Maps for subject 9.

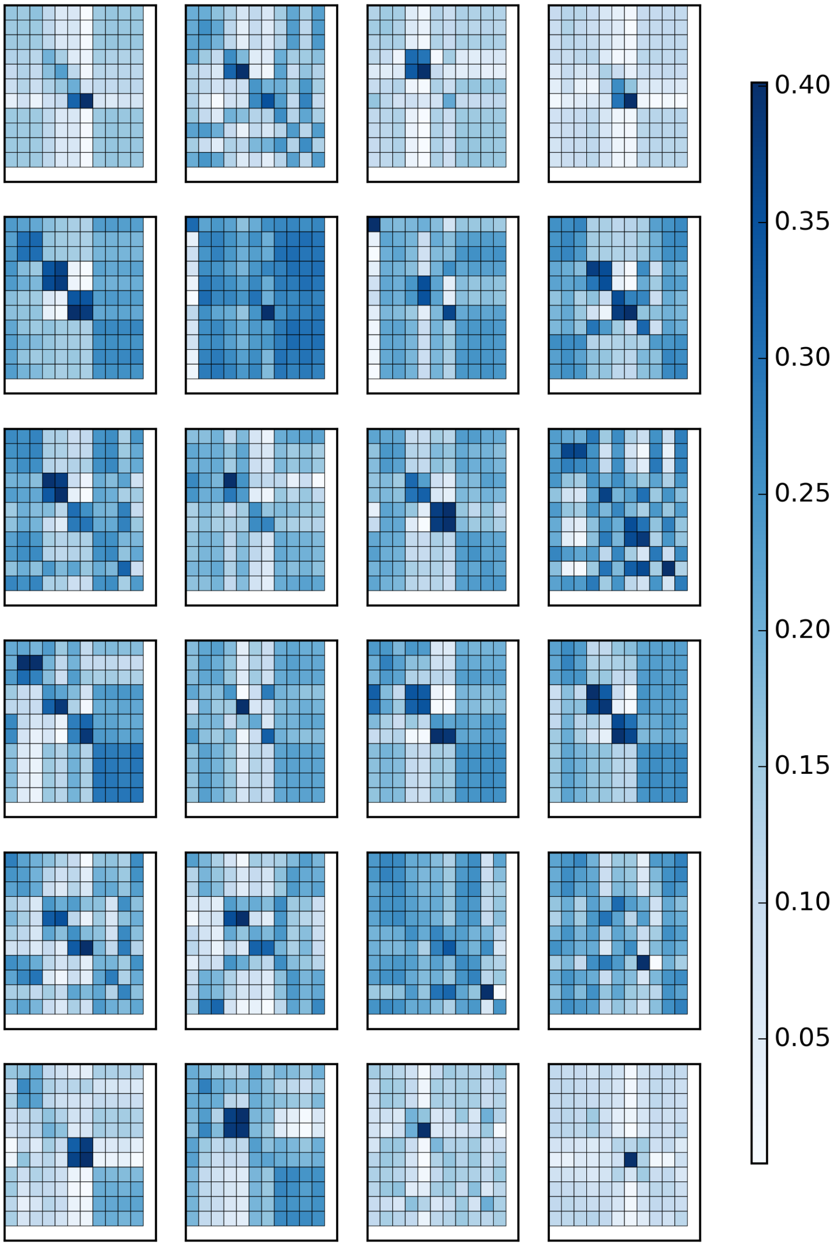

**Figure B.10**: Individual Probability Maps for subject 10.

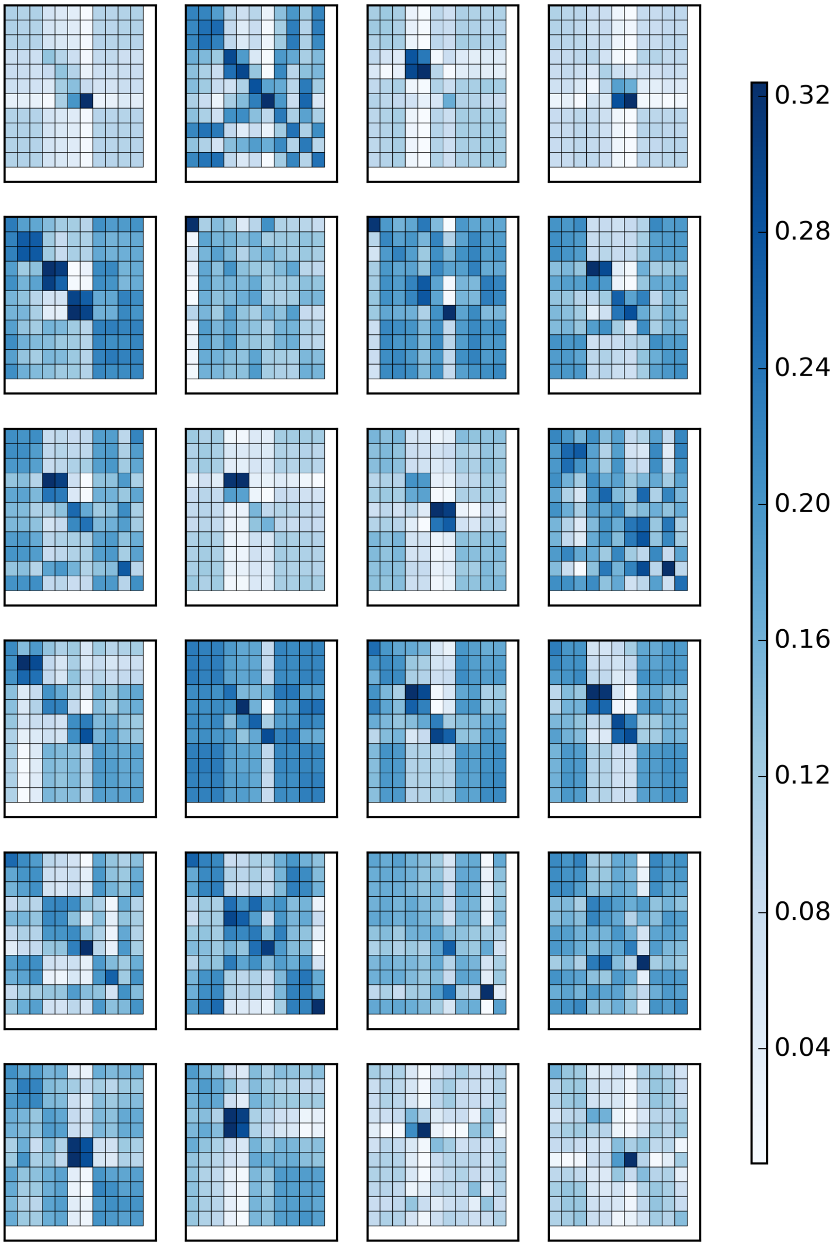

**Figure B.11**: Individual Probability Maps for subject 11.

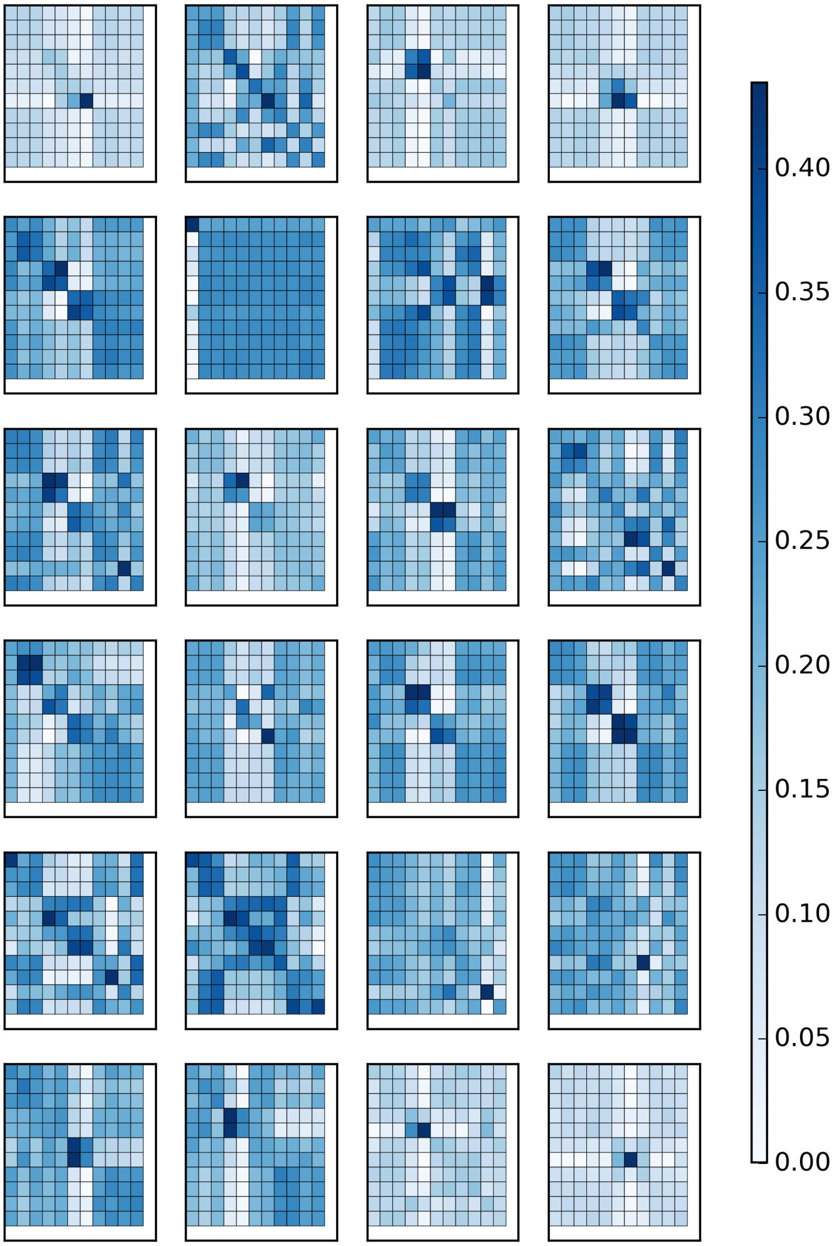

**Figure B.12**: Individual Probability Maps for subject 12.

# APPENDIX C

# PROGRESSION PROBABILITY MAPS

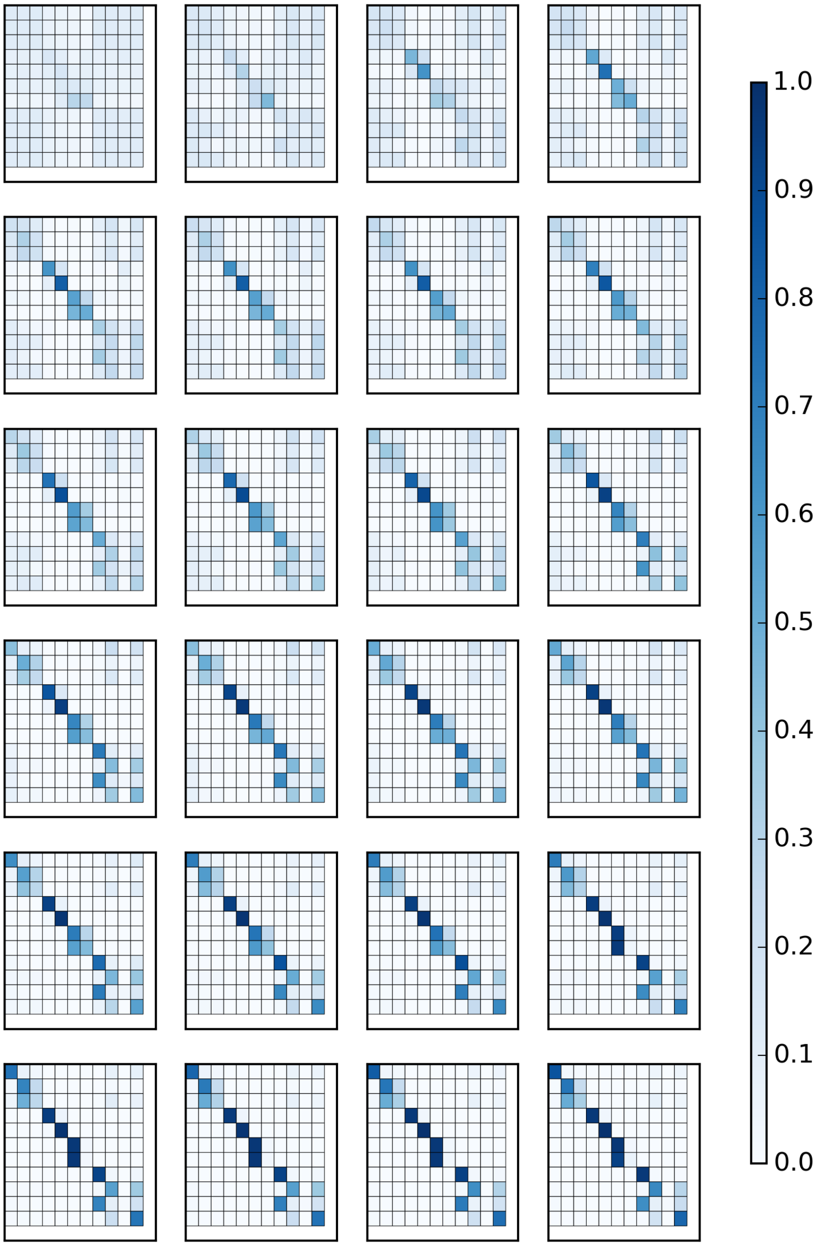

**Figure C.1**: Progression of Probability Maps for subject 1.

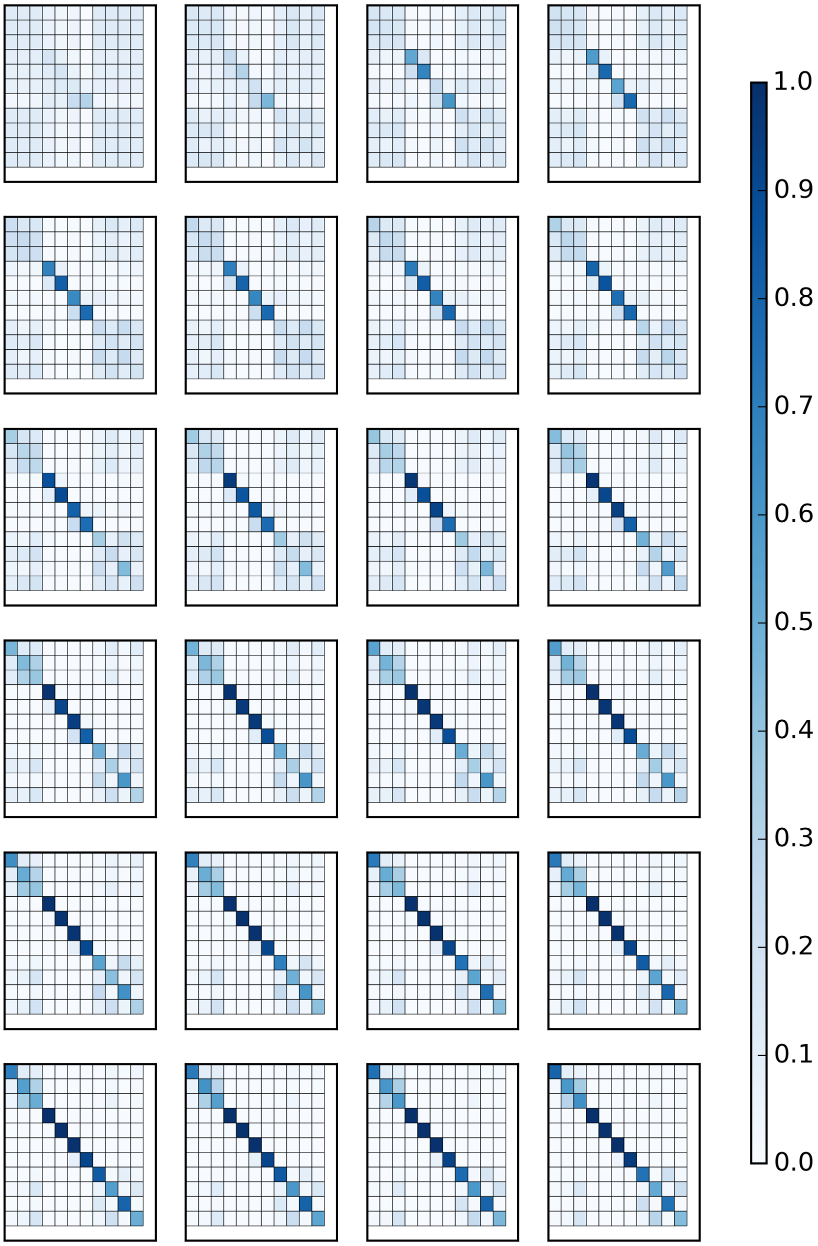

**Figure C.2**: Progression of Probability Maps for subject 2.

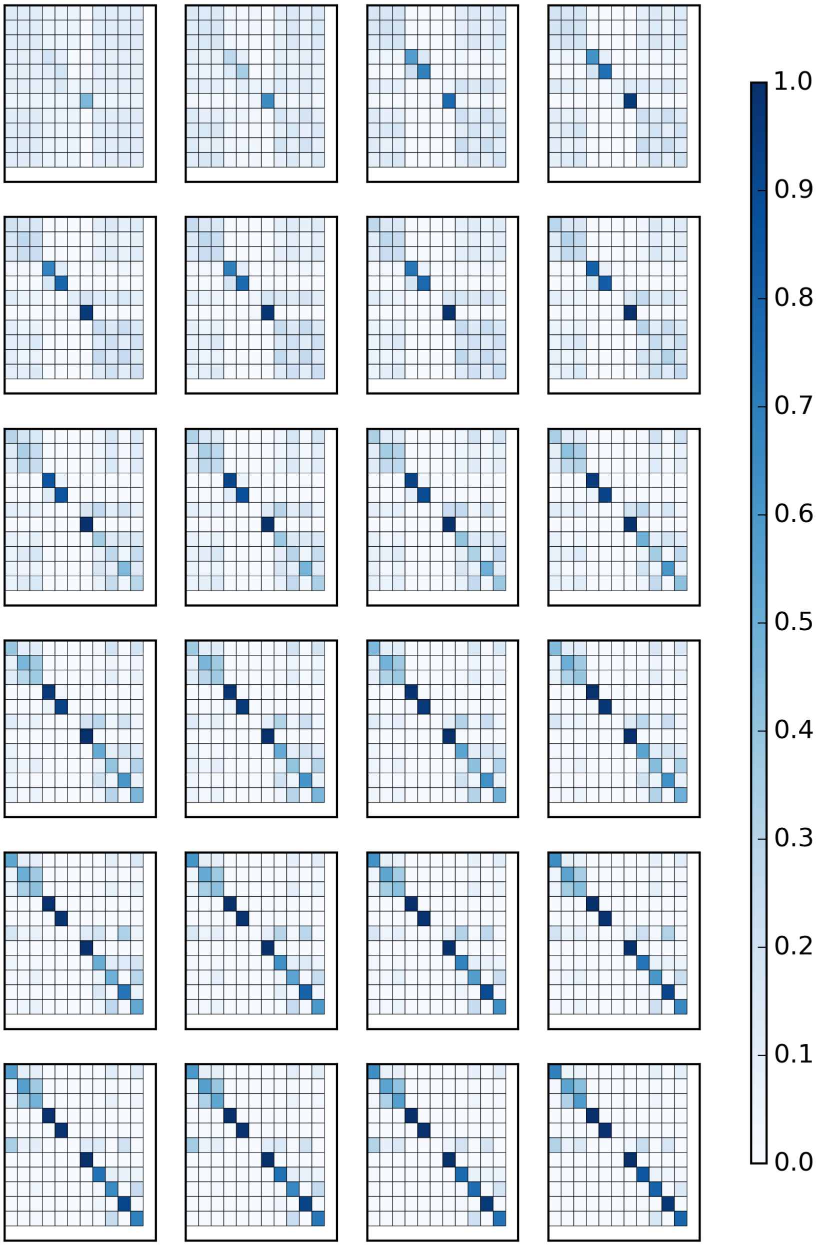

**Figure C.3**: Progression of Probability Maps for subject 3.

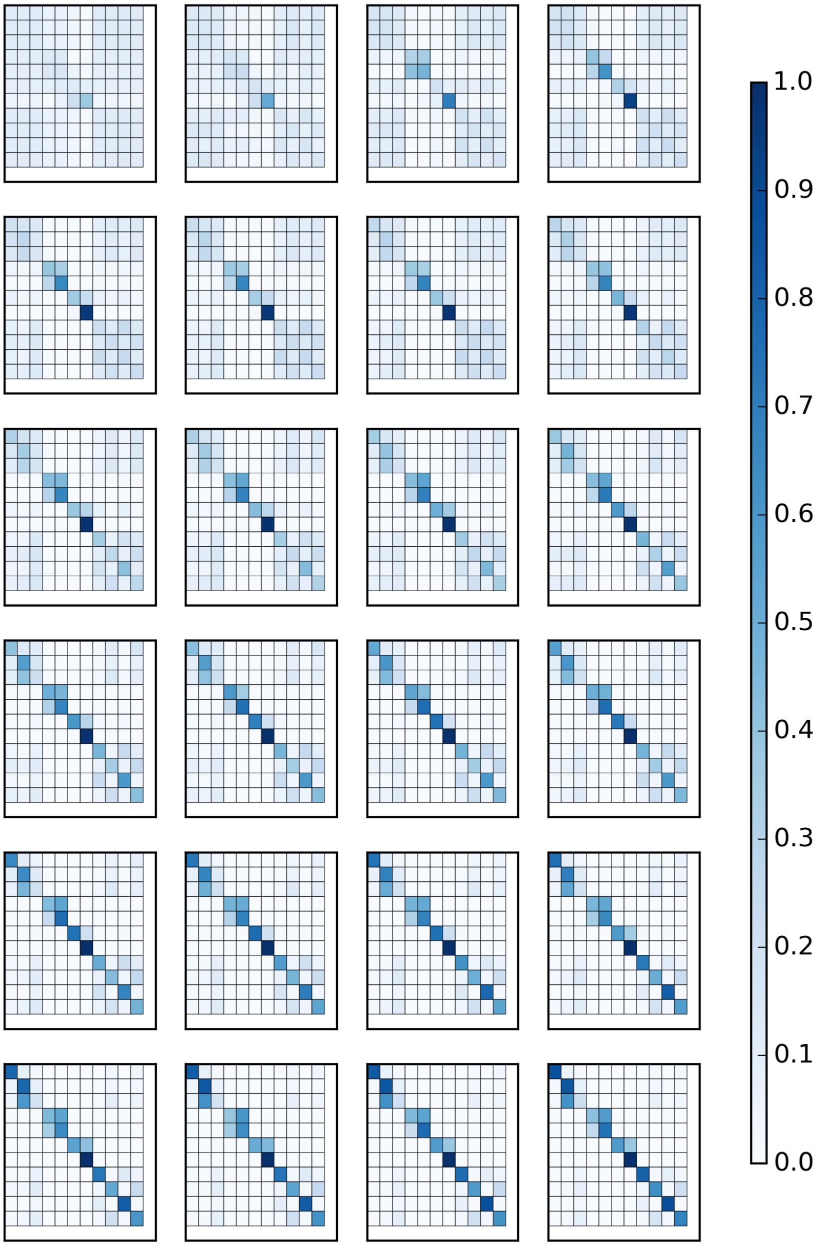

**Figure C.4**: Progression of Probability Maps for subject 4.

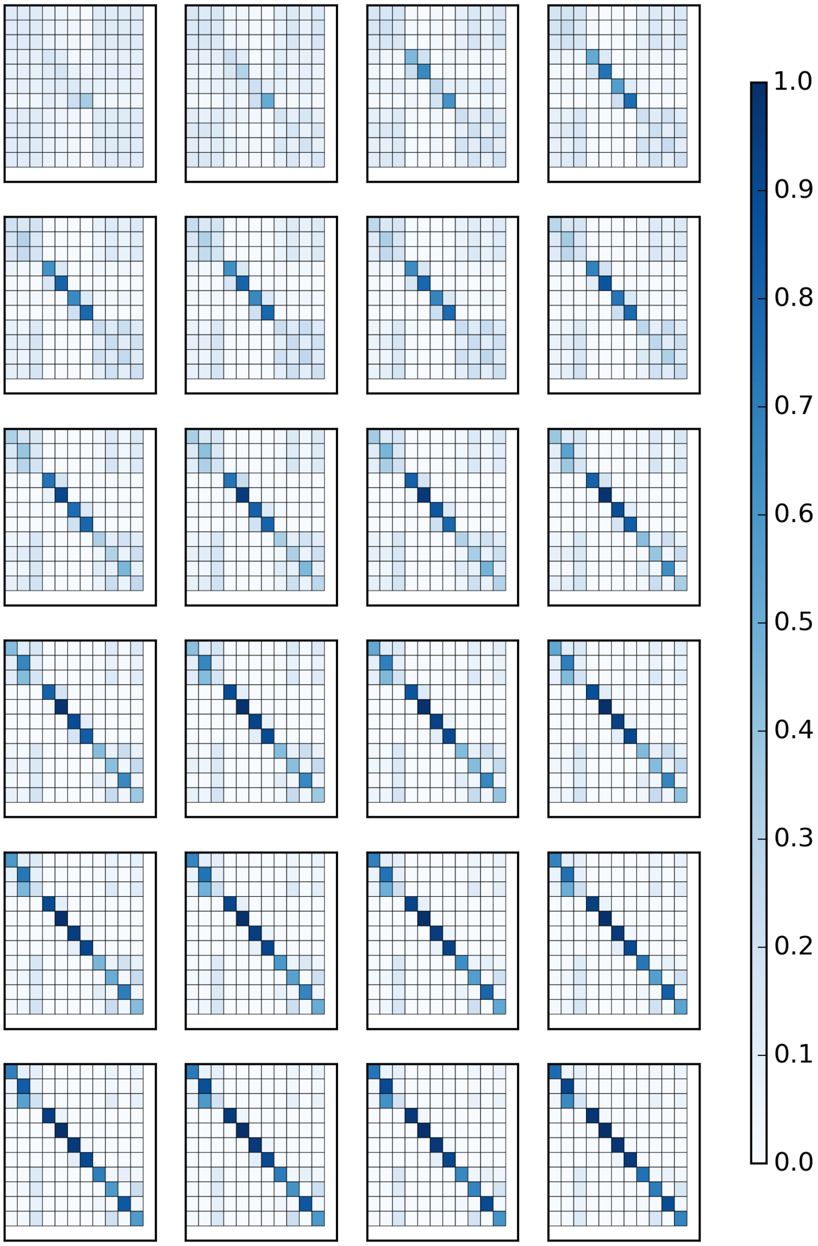

**Figure C.5**: Progression of Probability Maps for subject 5.

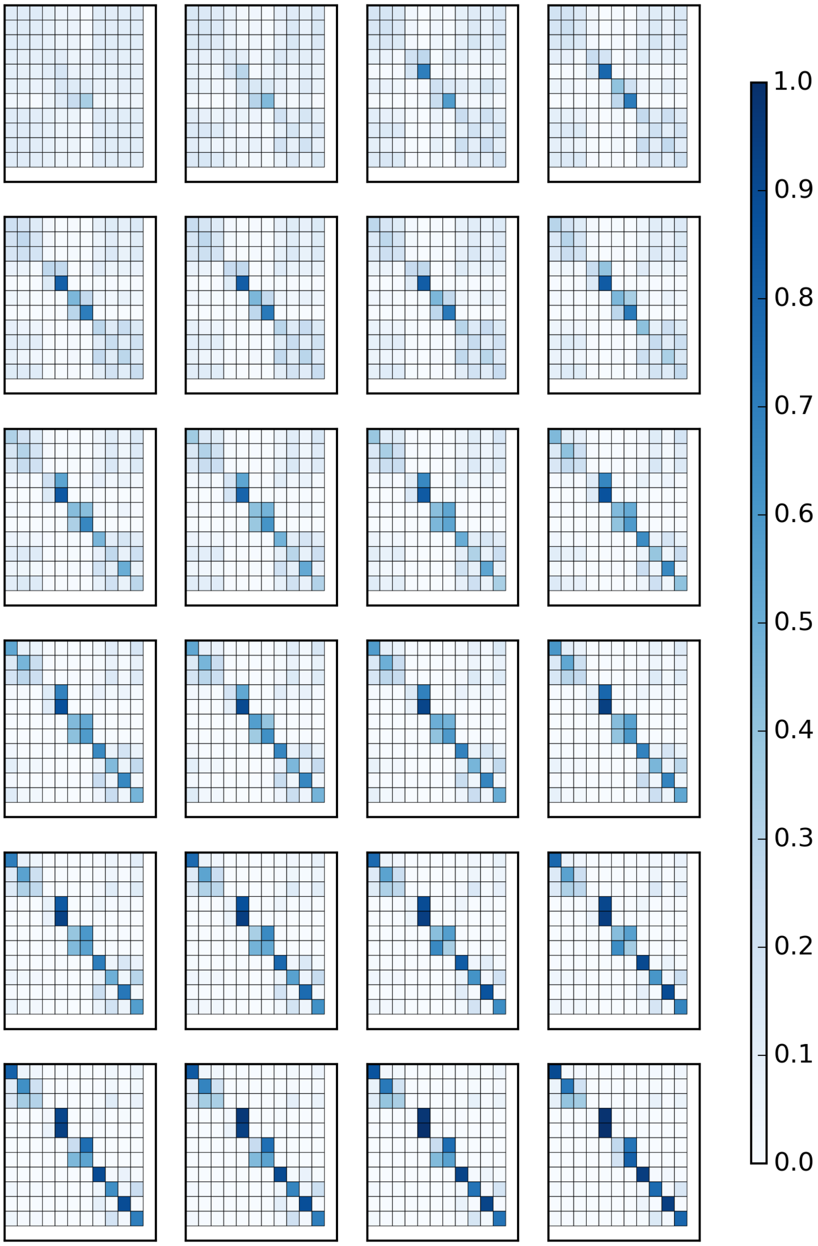

**Figure C.6**: Progression of Probability Maps for subject 6.

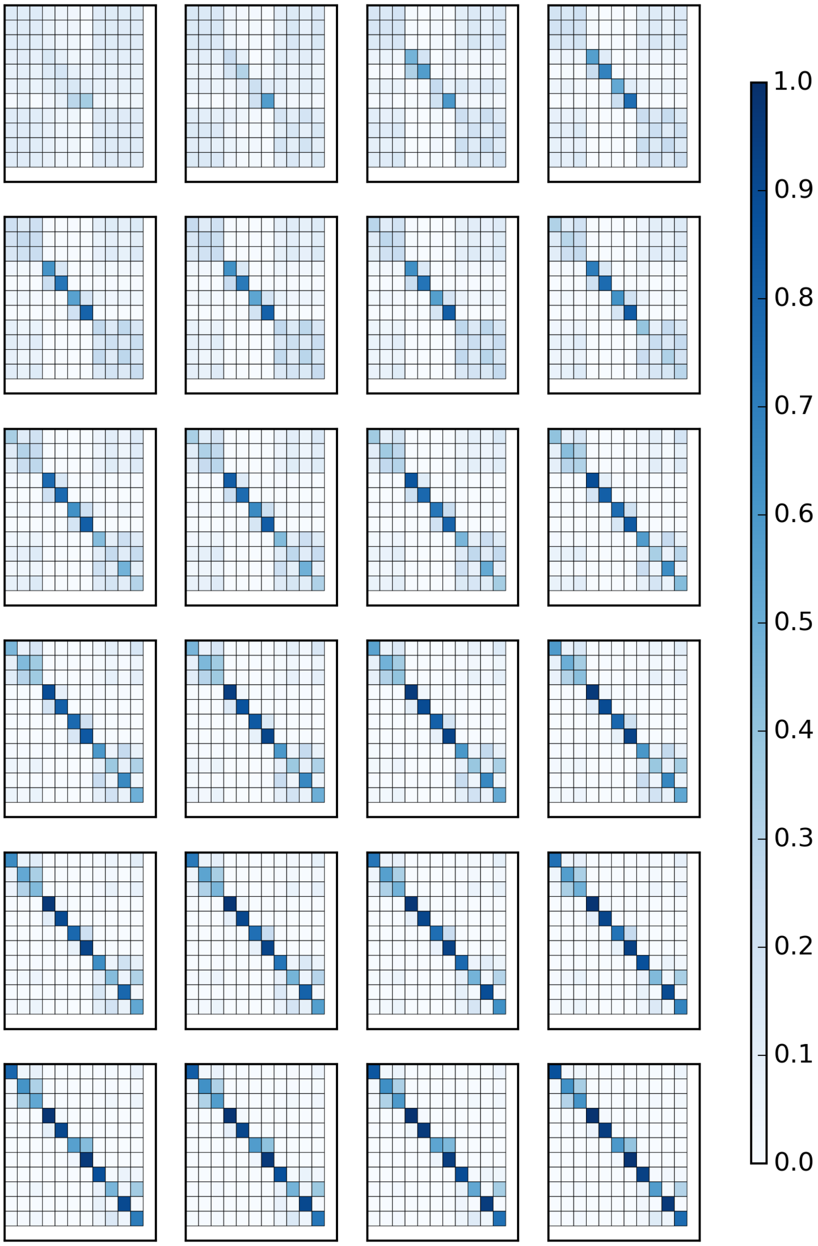

**Figure C.7**: Progression of Probability Maps for subject 7.

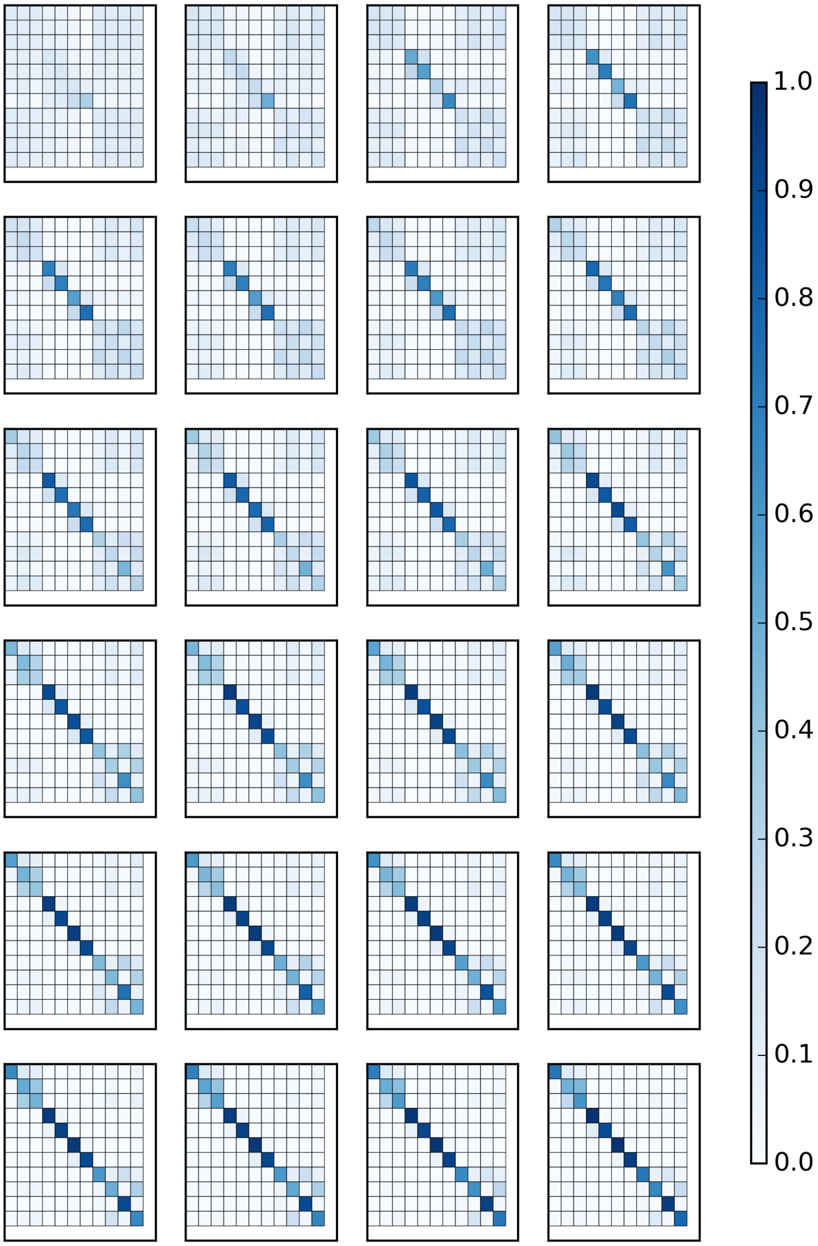

**Figure C.8**: Progression of Probability Maps for subject 8.

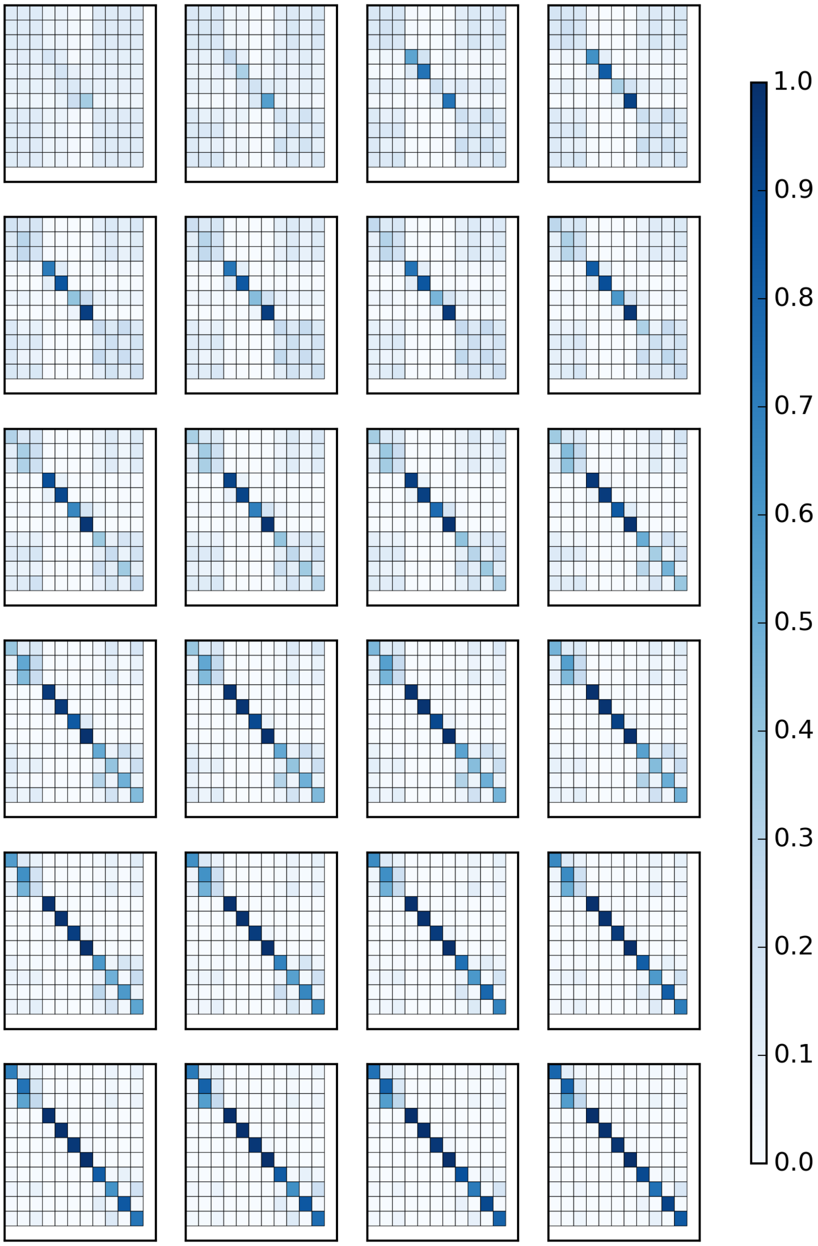

**Figure C.9**: Progression of Probability Maps for subject 9.

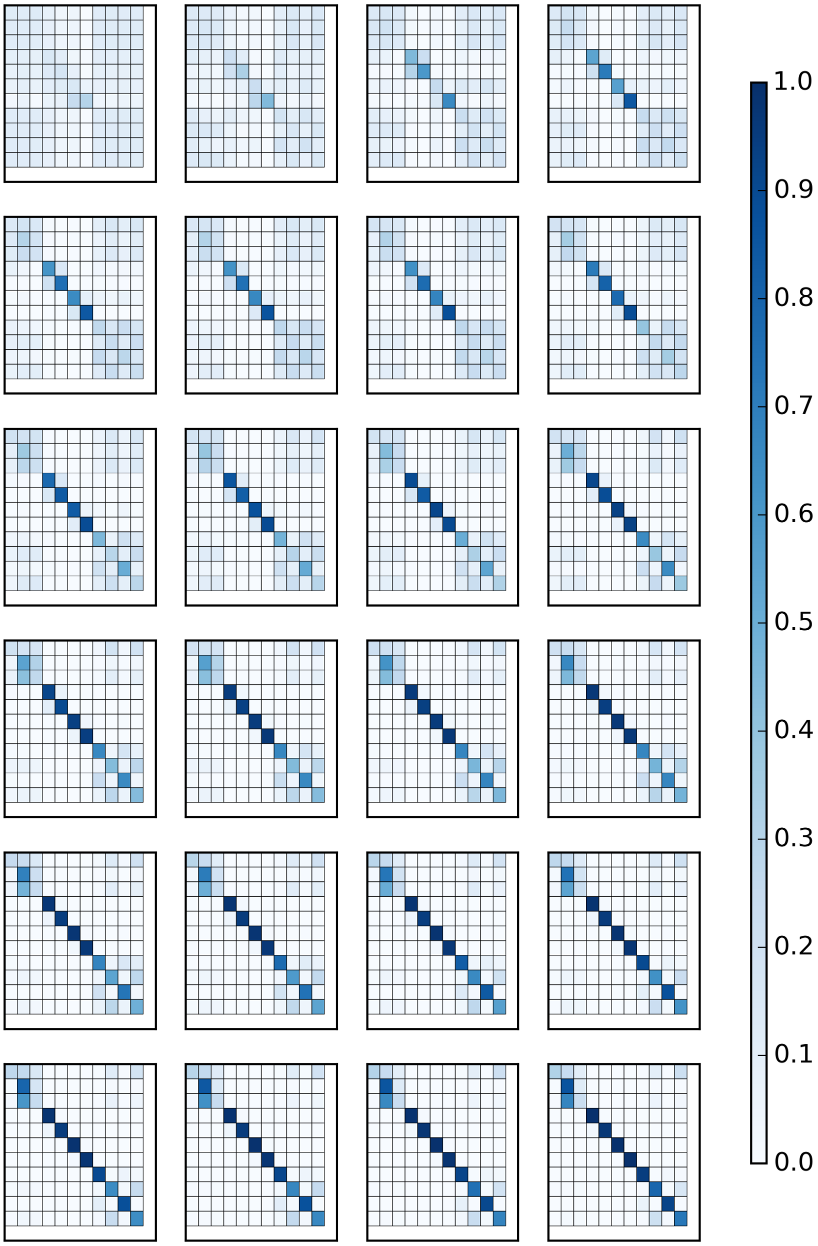

**Figure C.10**: Progression of Probability Maps for subject 10.

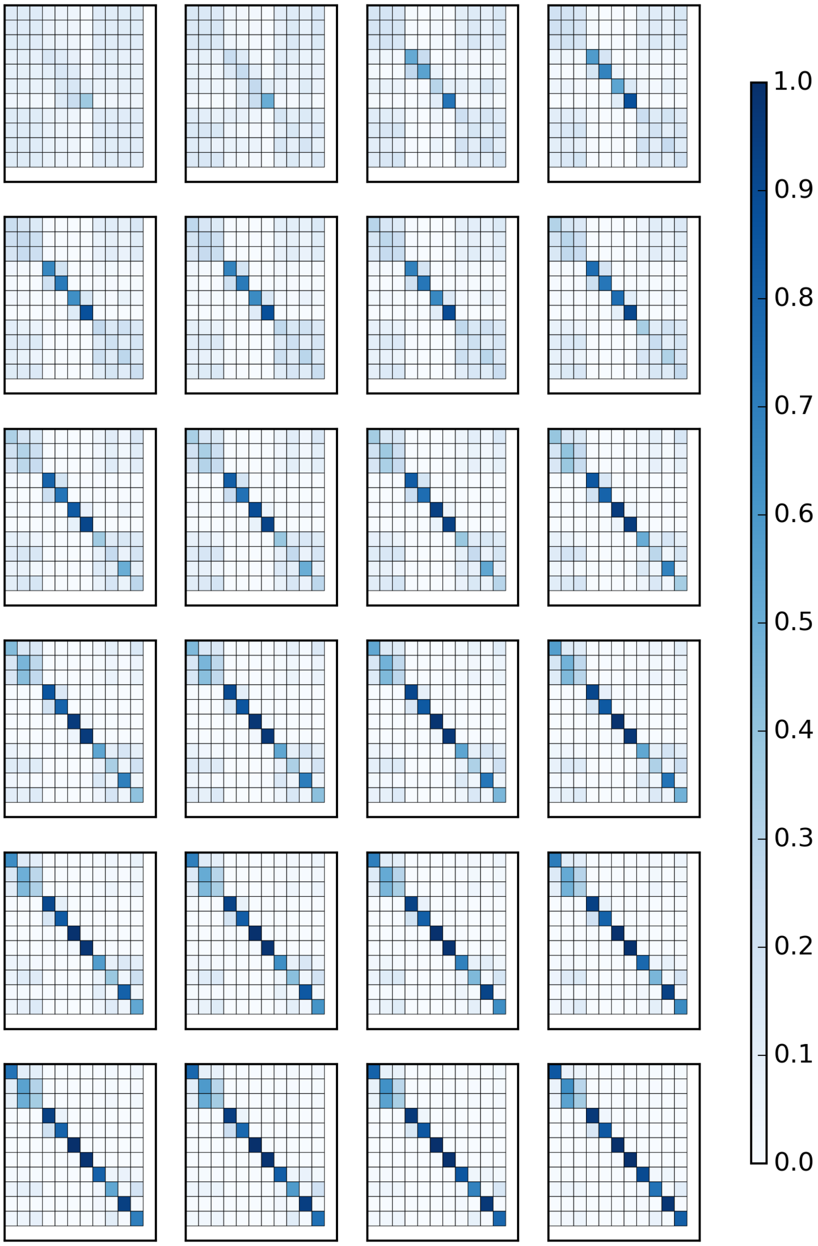

**Figure C.11**: Progression of Probability Maps for subject 11.

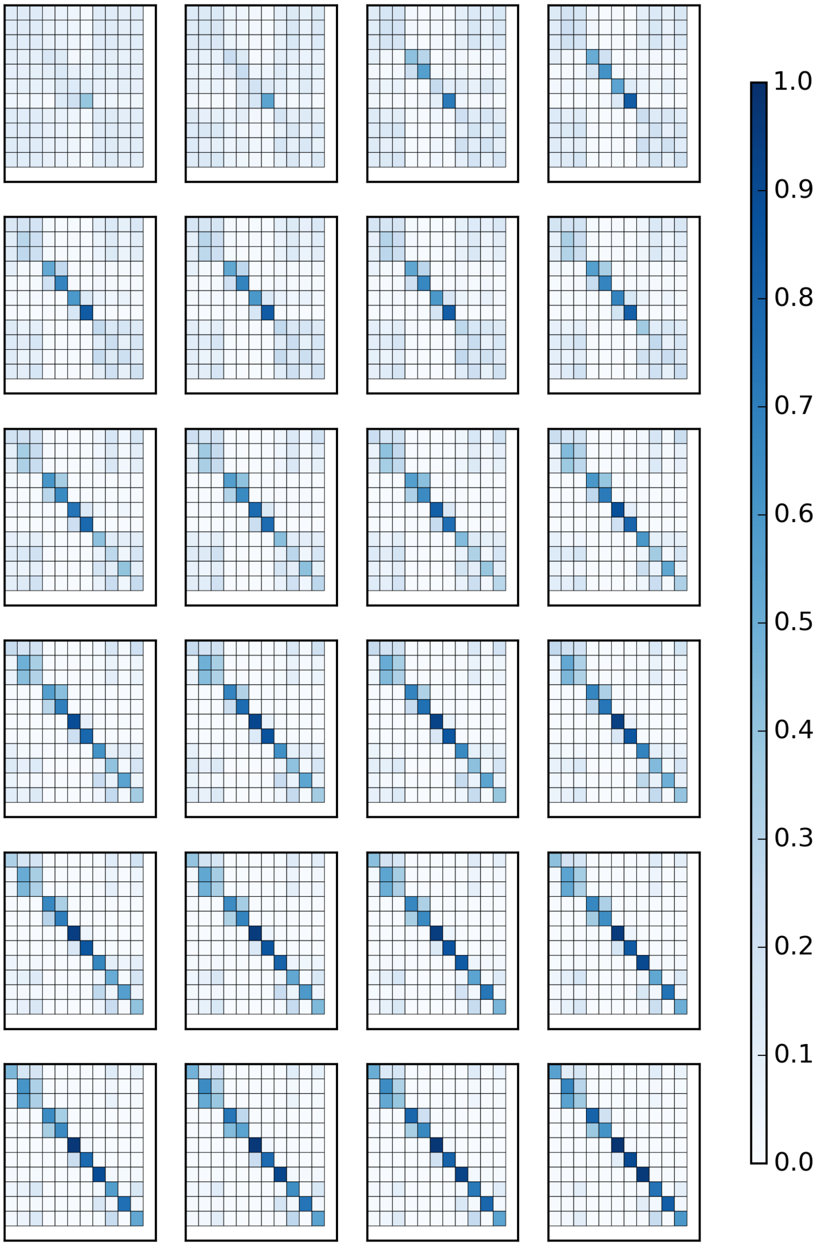

**Figure C.12**: Progression of Probability Maps for subject 12.