Sparton Navigation Modules
Integration and Calibration Guide

Applicable to Sparton AHRS-M2.
REVISION HISTORY:

<table>
<thead>
<tr>
<th>DATE</th>
<th>REV</th>
<th>DESCRIPTION OF CHANGE</th>
<th>INITIALS</th>
</tr>
</thead>
<tbody>
<tr>
<td>06/28/16</td>
<td>A</td>
<td>AHRS-M2 and AdaptCal</td>
<td>RSW</td>
</tr>
</tbody>
</table>
Table of Contents

1 Introduction ........................................................................................................................................... 4
  1.1 Scope................................................................................................................................................... 4
  1.2 References ........................................................................................................................................... 4
  1.3 Applicability ........................................................................................................................................ 4

2 Overview .................................................................................................................................................. 4
  2.1 Protocol Introduction ........................................................................................................................... 4

3 How do I setup the sensor ....................................................................................................................... 5
  3.1 Where should I place the sensor inside my Device ............................................................................. 5

4 Virtual Alignment Procedures .............................................................................................................. 5
  4.1 AHRS-M2 ............................................................................................................................................ 6
    4.1.1 Tare Procedure using North Reference ......................................................................................... 6
    4.1.2 Tare Procedure using a Known Azimuth, Pitch and Roll ............................................................. 7
    4.1.3 BoresightMatrix Adjustment ....................................................................................................... 10

5 In-Field Calibration – Procedures and Protocols .................................................................................. 10
  5.1 AHRS-M2 .......................................................................................................................................... 11
    5.1.1 Procedures .................................................................................................................................... 11
    5.1.2 Protocols ....................................................................................................................................... 11
      5.1.2.1 NorthTek ................................................................................................................................. 11

A Examples of AdaptCal and manual calibration ..................................................................................... 16
  A.1 AHRS-M2 .......................................................................................................................................... 16
    A.1.1 AdaptCal calibration ..................................................................................................................... 16
      A.1.1.1 Example Adaptive Calibration ................................................................................................ 16
    A.1.2 Manual calibration ......................................................................................................................... 16
      A.1.2.1 Manual calibration with Inverted points. ............................................................................... 16
      A.1.2.2 Manual calibration with non-inverted points ............................................................................ 16

B How and When to use the In-Field Calibration related variables .......................................................... 18
  B.1 magFieldCalErr .................................................................................................................................. 18
  B.2 yawErrEst ............................................................................................................................................. 18
  B.3 absoluteYawErr .................................................................................................................................. 18
1 Introduction

This document is intended to aid a user in the integration and installation of the Sparton inertial sensor. This document outlines procedures for that purpose. This document applies to the AHRS-M2 device. In this document, these devices will be referred to as inertial systems and all references shall apply to all devices except where noted.

1.1 Scope

This document covers the following topics:

- In-field Calibration of the Sparton Inertial Systems
- Virtual Mechanical Alignment of the Sparton Inertial Systems

This document is not a hardware or mechanical data sheet.

1.2 References

This document is an umbrella document in that some of the interface descriptions for the inertial systems are described in external documentation. This document describes the relationship of these documents. The additional documentation needed to complete the entire interface description for the inertial system family is listed below.

- NorthTek System, Programming Manual

1.3 Applicability

This document applies to the Sparton AHRS-M2 Inertial Systems.

2 Overview

2.1 Protocol Introduction

The interface to the inertial system is a serial communication link (User Port). The link is an asynchronous character oriented interface that operates in a bidirectional mode. This serial link transmits and receives asynchronous characters using nominal “UART” type framing with 8 data bits, no parity and a single stop bit, with a default baud rate of 115200. The baud rate may be changed by software command. No error detection is used with the individual characters.

The serial link is full-duplex; however in practice the information flow is largely half-duplex in nature. Essentially the client and the server operate in a poll-response fashion. For example, in the above
situation, the device that desires to use the inertial system would be the master (client) and the inertial system would be the slave (server).

3 How do I setup the sensor

Sparton Inertial Systems do not require typical setup to work but we do have evaluation software that requires installation. Each environment is different so there can be tweaks made to ensure the best possible accuracy if needed.

3.1 Where should I place the sensor inside my Device

Due to the nature of the Sparton Inertial System, placement of the sensor inside the host device may not be as easy as finding the space to fit. When selecting the location, it is necessary to find a spot where the magnetic disturbance is at a minimum.

To characterize the host system and to find a good location for placement of the Inertial System, it is recommended to take data from the sensor at different positions inside the host device. During this profile you will be determining the magnetic signature of the host device and its components. The following procedure describes how to collect the data for one position:

1. Prior to data logging please log the output of the following commands and save to a text file and the reset the device by removing power or sending the ‘reset’ command:
   
   ```
   save_mask d.on<cr>
db.print<cr>
   ```

2. Send the following commands and log the data:
   a. AHRS-M2
      
      ```
      magp_mask d.on<cr>- Sending/Pressing CTRL-q and CTRL-S will start/stop the output.
      ```

      What you are looking for is that the data has a low standard deviation/variance and that the DC offset, if any, is static and does not drift over time. If the DC offset of the magnetic signature is time-varying, then the previous In-Field Calibration will no longer apply.

3. This dataset should be a profile of the host targeting system. Take care to note what is being done during each log. Turn on/off individual components if possible: camera zoom, camera focus, GPS acquiring/tracking. This will allow a user to know at which times the output is more accurate and which internal systems will affect the magnetometer readings.

4 Virtual Alignment Procedures

For an accurate measurement it is imperative that the Sparton Inertial System be aligned properly with the host system; in other words, to be pointing towards where the end device is pointing.
Sparton has the ability to adjust the Rotation Matrix (boresightMatrix) on the fly. This allows the sensor to be mounted into a near infinite number of orientations and be virtually rotated to match the host system. While the gyros do not need to be at the center of rotation for each axis, the accelerometers will experience centripetal accelerations which will affect the pitch/roll and effectively the heading. In static environments there is less of a concern with this issue.

Measurement due to misalignment is very problematic for some applications. An example of how misalignment error affects heading is as follows:

<table>
<thead>
<tr>
<th>misalignment</th>
<th>pitch</th>
<th>roll</th>
<th>Yaw error</th>
</tr>
</thead>
<tbody>
<tr>
<td>10° Pitch / 10° Roll</td>
<td>0.05</td>
<td>0.05</td>
<td>0.017638</td>
</tr>
<tr>
<td>20° Pitch / 20° Roll</td>
<td>0.05</td>
<td>0.05</td>
<td>0.036418</td>
</tr>
<tr>
<td>30° Pitch / 30° Roll</td>
<td>0.05</td>
<td>0.05</td>
<td>0.057785</td>
</tr>
<tr>
<td>45° Pitch / 45° Roll</td>
<td>0.05</td>
<td>0.05</td>
<td>0.100124</td>
</tr>
<tr>
<td>60° Pitch / 60° Roll</td>
<td>0.05</td>
<td>0.05</td>
<td>0.173467</td>
</tr>
<tr>
<td>70° Pitch / 70° Roll</td>
<td>0.05</td>
<td>0.05</td>
<td>0.275197</td>
</tr>
</tbody>
</table>

### 4.1 AHRS-M2

#### 4.1.1 Tare Procedure using North Reference

This procedure is the simplest procedure but the user needs to know the North Direction to a high degree of accuracy without using the Sparton Device for Northing.

- Point the Host System to North and Zero degrees Pitch and Roll.
  - Do not use the Sparton sensor for this procedure as this procedure is meant to align the Sparton with the host system.
- Open the Spartacus Software.
- Go to Device Settings->Tare Device
  - This will finish in about one second and the Virtual Alignment Procedure will be complete
- If you are using a terminal emulator or a microcontroller, after the system has been pointed to North and leveled you would send the command `InvokeTare 0 set drop` to run the tare procedure.
### 4.1.2 Tare Procedure using a Known Azimuth, Pitch and Roll

**When using scripts this large you must set up the system as follows:**

- Insert a delay per line of 5ms.
- Ensure each line “to be processed” (not comment lines) is less than 60 characters (bytes).

This procedure was created due to some customers not knowing North Direction but knowing another azimuth with a high degree of accuracy. The best results will occur if the roll is kept at 0.0°. Any error in the pitch or roll of the host system will compound as describe in the previous section.

- Point the Host System to a known azimuth, pitch, and roll.
- Open the Spartacus.
- Go to File-> Send NorthTek Script
- Browse to the location of the script (apr_tare.4th) and select the file.
- Click the Open button.
- Scan the terminal window to make sure the word "Huh?" does not show. This will allow us to make sure the script loaded properly.
- Type in the command with the know azimuth, pitch, and roll as follows:
  - For azimuth/pitch/roll of 191.4/49.5/0.0 type "f191.4 f49.5 f0.0 apr_tare<enter>"
  - The Boresight Matrix will print out and the operation will be complete

```plaintext
// If the script gets reloaded in the same session
// Forget the previous version
// Needs revision 2.1.1 or later.
// This forgets the comp_rot_matrix from a previous load of the macro
// this is standard practice with NorthTek for debugging
// so as to not overflow the wordlist space.
// When this file is reloaded with a terminal a second time
// this removes the previous program.
forget comp_rot_matrix

// Declare a few working variables.
variable comp_rot_matrix 9 allot
variable sat_rot_matrix 9 allot
variable tare_rot_matrix 9 allot
variable azimuth
variable pitch1
variable roll1
variable cosine_phi
variable sine_phi
variable cosine_theta
variable sine_theta
variable cosine_psi
variable sine_psi
variable copyarray 5 allot

// Function (aka NorthTek word): index!
```
// Shorthand to compute an array index into the variable
// copyarray and write the given value to that index position
// Inputs:
// TOS-1: value to be written
// TOS: index into copyarray
// Modifies:
// NorthTek variable: copyarray
// This word takes a value and an index and stores that element
// in that position in the copyarray.
//******************************************************************************
(value index -- )
index! 4 * copyarray + !;

// Function (aka NorthTek word): cp
(array_ptr -- )
// copies a 3 element array into the copyarray
// copyarray ends up with 0 2 V1 V2 V3 which is
// the right form for setting in the database
// Inputs:
// TOS: ptr to source array
// Modifies:
// NorthTek variable: copyarray
// ********************
: cp
0 0 index! // store 0 in position 0 of copyarray
2 1 index! // 2 in position 1, therefore we set items 0..2
dup @ 2 index! // make copy of pointer, for next index, store 1st element.
4 + dup @ 3 index! // move to second element, make copy, store value
4 + @ 4 index! // move to third element, store it.
;
: compute_vars
roll1 @ d>r cos cosine_phi !
roll1 @ d>r sin sine_phi !
pitch1 @ d>r cos cosine_theta !
pitch1 @ d>r sin sine_theta !
azimuth @ d>r cos sine_psi !
azimuth @ d>r sin sine_psi !
;
// Functions (aka NorthTek words): row0, row1, row2
// some convenient shorthand for matrices
// Inputs:
// TOS: Ptr to start of a matrix
// Outputs:
// TOS: Ptr. to start of a row within the given matrix
//******************************************************************************
// Since in Forth you must do matrix/array index calculations
// explicitly, these operators just make it easy to
// get to rows 1 and 2 of a 3x3 matrix.
: row0 0 + ;
: row1 12 + ;
: row2 24 + ;

// Function (aka NorthTek word): compute
// The actual computation
{ -- }
// This stack diagram indicates that there are no params
// and no explicit results.
// This function uses the global variables comp_rot_matrix and sat_rot_matrix
// Uses:
// Database variables: cp2, cp1, accelEst
// Modifies:
// NorthTek variable: comp_rot_matrix
//******************************************************************************
: compute
compute_vars
cp2 di8 cp1 di8 accelEst di8 // get the three desired columns
```cpp
comp_rot_matrix buildMatrix            // build the comp_rot_matrix with rows
    cosine_theta @ cosine_psi @ f* sat_rot_matrix row0 !
    cosine_theta @ sine_psi @ f* sat_rot_matrix row0 4 + !
    f0.0 sine_theta @ f- sat_rot_matrix row0 8 + !

    cosine_psi @ sine_theta @ sine_phi @ f* f* cosine_phi @ sine_psi @ f* f+
    sat_rot_matrix row0 !
    sine_psi @ sine_theta @ sine_phi @ f* f* cosine_phi @ sine_psi @ f* f+
    sat_rot_matrix row1 4 + !
    cosine_theta @ sine_phi @ f* sat_rot_matrix row1 8 + !
    cosine_phi @ sine_theta @ sine_psi @ f* f* cosine_psi @ sine_psi @ f* f+
    sat_rot_matrix row2 !
    sine_psi @ sine_theta @ sine_psi @ f* f* cosine_phi @ sine_psi @ f* f+
    sat_rot_matrix row2 4 + !
    cosine_theta @ cosine_psi @ f* sat_rot_matrix row2 8 + !
    cosine_psi @ cosine_psi @ f* sine_psi @ cosine_psi @ f* f-
    sat_rot_matrix m*m>r ;

    // ********
    // ***************
    // Function (aka NorthTek word): copyit
    // Copy each row of the matrix into the
    // The corresponding row of the boresight matrix.
    // Uses:
    // NorthTek variable: matrix
    // Modifies:
    // Database variables:
    // boresightMatrix
    // *****************************************************
    : copyit
    tare_rot_matrix boresightMatrix m>m

    // ***************
    // Function (aka NorthTek word): printit
    // Printout the computed tare_rot_matrix and the current boresight
    // matrix.
    // *****************************************************
    : printit
    tare_rot_matrix cr m.       // Use the matrix print function
    boresightMatrix di.        // Use the database print function

    // ****************
    // Function (aka NorthTek word): tare
    // Create a small program to perform the Tare function
    // Uses:
    // Database variables (used by compute): cp2, cp1, accelEst
    // Modifies:
    // NorthTek variable: tare_rot_matrix
    // Database variable: orientation
    // Database variables (set by copyit):
    // boresightMatrix
    // *****************************************************
    : apr_tare
    roll1 !
    pitch1 !
    azimuth !
    tare_rot_matrix clear(m)     // clear the matrix we declared
    orientation 0 set            // setup to default orientation, required.
    1000 delay                   // Wait 1 second for this to settle in computation
    compute copyit               // compute the matrix and copy to the boresight matrix.
    printit                      // Print it for verification.
;
    // Input format is azimuth pitch1 roll1 apr_tare CR
    // ex. f191.4 f49.5 f15.0 apr_tare [CR]
```
4.1.3 BoresightMatrix Adjustment

This procedure is the most usable as it can adjust a boresightMatrix to any predefined orientation the user wishes.

Example is to find the Direct Cosine Matrix of rotation and implement using the following way:

Example:

\[
\begin{bmatrix}
0 & 0 & 2.2 \\
1.000000e+00 & 0.000000e+00 & 0.000000e+00 \\
0.000000e+00 & 1.000000e+00 & 0.000000e+00 \\
0.000000e+00 & 0.000000e+00 & 1.000000e+00 \\
\end{bmatrix}
\]

\text{boresightMatrix =}

\[
\begin{bmatrix}
0.000000e+00 & 0.000000e+00 & 0.000000e+00 \\
0.000000e+00 & 1.000000e+00 & 0.000000e+00 \\
0.000000e+00 & 0.000000e+00 & 1.000000e+00 \\
\end{bmatrix}
\]

5 In-Field Calibration – Procedures and Protocols

In the presence of Hard and Soft Iron magnetic distortions, the device must be calibrated for use in each specific environment. For a hard or soft iron distortion that exceeds the Earth’s local magnetic field, the resolution of the heading will be reduced. The greater the field strength of the offending part, the less resolution and accuracy of the heading.

To explain more fully, this is a platform calibration. Static distortions that are rotating together with the Sparton Inertial System can be calibrated out provided that they are not saturating the magnetic field sensors. Only those distortions that have enough magnetic field strength to be detected by the sensors need to be calibrated out. Since we cannot “see” the magnetic field strengths of any material, you must allow the Sparton Sensors to “show” you what is affecting them. An easy way to perform this task is to log or watch the magnetic sensor readings while moving items in your device closer to the system and then slowly moving them away until the magnetic effects detected by the sensors are removed.
5.1 AHRS-M2

*After the first calibration point is taken*, the previous magnetic values will be removed from the magnetic-based yaw/heading calculation. This will show up as a heading drift in the output. Since the magnetometer values are being calibrated at this point, it is assumed that these previous values are not correct. To alleviate this drift, you can do one of the following:

1. Reset the magnetometers back to their original value before performing the in-field calibration. To do this, modify the following variables prior to beginning the in-field calibration.
   a. `clearPointCal 1 set drop` or `clearFieldCal 1 set drop`
2. Turn off the magnetometers and rely only on the accelerometers and gyros during the calibration point selection. Then turn them back on once the calibration is complete.
   a. `magAiding 0 set drop` to turn them off.
   b. After calibration, reset them to ON by `magAiding 1 set drop`

In the following procedures, we describe how to calibrate out those distortions that cannot, for whatever reason, be removed from the host device.

5.1.1 Procedures

Field calibration involves multiple steps:

1) Select which mode: Manual Calibration or AdaptCal.
2) Start the calibration
3) Capture Points
4) End point capture
5) Allow the calibration algorithm to converge.
6) Terminate calibration, causing values to be stored.

The compass is already factory calibrated in an environment free from magnetic distortions. When the compass is first used in the application, it must learn the local magnetic distortions.

5.1.2 Protocols

5.1.2.1 NorthTek

NorthTek is not a protocol per se. NorthTek is a programming language, a command interpreter, and an execution environment. NorthTek is described in detail in a separate manual. NorthTek provides the user with a command line interface that allows direct interaction with the inertial system. NorthTek also allows the user to load custom programs into the inertial system that will execute a custom user application. The user may create programs that cause some of the standard protocol outputs but at user defined points, or may create custom output depending on the specific need. The user specific algorithms, for example, may be used to filter the output data, control the reporting rate, create unusual mounting configurations, or select multiple calibration sets.
Because NorthTek is an environment unto itself, an entire manual is dedicated to the NorthTek System as listed in the references. The user should refer to that manual for detailed descriptions of the commands being used in this manual. You can see each of the appropriate delays in the NorthTek scripts provided in this document regarding the in-field calibration procedure.

NorthTek provides command line access to the internal database variables. NorthTek also provides raw and processed sensor data streamed at the acquisition rate. The sections that follow illustrate some of the NorthTek functionality that can be used on the inertial system. The reader should refer to the NorthTek System Programming Manual for detailed descriptions on syntax and semantics of the commands used in the examples that follow.

Due to the critical nature of the in-field calibration procedure the recommended delays between commands are 250-300ms.

5.1.2.1.1 AdaptCal Compass Calibration

When using scripts this large you must set up the system as follows:
- Insert a delay per line of 5ms.
- Ensure each line “to be processed” (not comment lines) is less than 60 characters (bytes).

This script performs the adaptive in-field compass calibration. To perform calibration use a terminal emulator to send this to the Inertial system (if huh? is seen flying by, add some delay to each line transmitted or reduce the baud to 38400). Once the script has been loaded, send the command “adaptCal”. The script will then prompt the user to capture calibration points. The user should move the compass around and select between 4 and 12 points, then hit ESC. The script will then print the magnetic error at a 0.5 Hz rate. Observe the magnetic error until it converges sufficiently then hit the spacebar or any other key. The compass will now be field calibrated. The “adaptCal” command may be re-entered as many times as desired without re-loading the macro until the unit is shut off. The macro needs to be loaded only once while the inertial system remains powered up. There is enough information in the basic programming section and the appendix to analyze this macro sufficiently should the user desire to extend this functionality to other custom user scripts.

```
// ************************************************************
// NorthTek Script for adaptive calibration.
// ************************************************************
// This erases this script should it be reloaded.
forget adaptCal
  // User command to clear the point cal results
  clearPointCal 1 set drop
  300 delay
```
// User command to clear the adaptive cal results
clearFieldCal 1 set drop
300 delay

// Initialize the mag and accel calibration point buffer to all zeros
initCalPointBuffer 1 set drop
300 delay

// begin the adaptive calibration mode
autoFieldCalActive 1 set drop

." Calibration starting" cr
350 delay

." Press any key to terminate" cr

// This loop till continue to print the: (timing is print every ~0.5 seconds)
// 1) number of points in the buffer used for calibration
// 2) The calibration score in possible degrees of heading error (magFieldCalErr)
// 3) The actual yaw error estimate which includes the calibration score
begin
  ?key 0=
  while
    magBufferActiveIndex di.
    magFieldCalErr di.
    yawErrEst di.
    490 delay
  repeat
  key drop

." Calibration done!" cr

// Turn off the Adaptive Calibration
// This is not necessary unless you specifically would like to turn off the
Adaptive Calibration.
// The AdaptCal is automatically being used once at least 4 points are collected.
// In this way, the AdaptCal and manual point cal differ
// The manual cal needs to be switched off for the calibration to be used.
autoFieldCalActive 0 set drop
350 delay

// Print out the calibration score one last time.
magFieldCalErr di.

5.1.2.1.2 Manual Point Compass Calibration

When using scripts this large you must set up the system as follows:

- Insert a delay per line of 5ms.
- Ensure each line “to be processed” (not comment lines) is less than 60 characters (bytes).

This script performs Manual compass calibration. To perform calibration use a terminal emulator to send this to the Inertial system (if Huh? is seen flying by, add some delay to each line transmitted or reduce the baud to 38400). Once the script has been loaded, send the command “cal3D”. The script will then prompt the user to capture calibration points. The user should move the compass around and select between 4 and 32 points, then hit ESC. The script will then print the magnetic calibration error and the yaw estimate error at a 0.5 Hz rate. Observe the magnetic calibration error until it converges sufficiently then hit the spacebar or any other key. The compass is now field calibrated. The “manCal”
command may be re-entered as many times as desired without re-loading the macro. The macro needs to be loaded only once while the inertial system remains powered up. There is enough information in the basic programming section and the appendix to analyze this macro sufficiently should the user desire to extend this functionality to other custom user scripts.

```
// NorthTek Script for ManCal calibration.
// ****************************************
// This erases this script should it be reloaded.

forget clearCals

// User command to clear the point cal results
// User command to default to the point cal
// Initialize the mag and accel calibration point buffer to all zeros

: clearCals
   // User command to clear the point cal results
   clearPointCal 1 set drop
   300 delay

   // User command to clear the adaptivecal results
   clearFieldCal 1 set drop
   300 delay

   // Initialize the mag and accel calibration point buffer to all zeros
   initCalPointBuffer 1 set drop
   300 delay
;

// This loop till continue to print the: (timing is print every ~0.5 seconds)
// 1) number of points in the buffer used for calibration
// 2) The calibration score in possible degrees of heading error (magFieldCalErr)
// 3) The actual yaw error estimate which includes the calibration score

: convergenceLoop
begin
   ?key 0=
   while
   490 delay
   repeat
   key drop
;

: manCal
   clearCals

   ." Calibration starting" cr

   pointFieldCalActive 1 set drop
   350 delay

   ." Press any key to take next point, ESC to finish" cr

// Wait until key is pressed and either take a manual point or stop point capture
// ESC key ends pint capture
// Any other key captures a point and prints out the number of points captured

begin
```
key 27 = 0=
while
  manualCalibrationPoint 1 set drop
  1000 delay
  magBufferActiveIndex di.
repeat
.
" Starting error settling" cr
." Press any key to terminate" cr
convergenceLoop
.
" Calibration done!" cr

// Turn off manualcalibration and print out calibration score.
// Depending on how the variable is set, the calibration process ends or
//    1) transitions to adaptivecalibration mode
//    2) or shuts off completely
pointFieldCalActive 0 set drop
350 delay // give compass time to process
magFieldCalErr di.
;
A Examples of AdaptCal and manual calibration

A.1 AHRS-M2

A.1.1 AdaptCal calibration
When performing adaptive calibration, the user should always do the following:

1. The more diverse points taken, the better the convergence will be.
2. If the magnetic environment is too non-linear (near a building or large magnetic source that does not rotate with the system), the point buffer will be flushed and the calibration will start from scratch.

A.1.1.1 Example Adaptive Calibration
Here is an example of AdaptCal:

1. Turn on AdaptCal
2. Pitch system down to -30°
3. Rotate slowly in a circle.
4. Pitch system down to -60° and rotate slowly in a circle.

You may add rolling the device when rotating slowly in a circle as well to possibly improve the calibration score.

A.1.2 Manual calibration
There are an infinite number of possible solutions for the Manual calibration and only a select few will be described here. These are only a few and can easily be modified to improve the diversity over the sphere.

A.1.2.1 Manual calibration with Inverted points.
Sparton’s Manual calibration algorithm is best used when the points are as diversely spread as possible for a given environment. Here is an example with inverted points.

1. Start by finding the maximum magnetometer X-Value in the X-Y plane and use this as the starting point
2. Take first point with Pitch increased to 45°.
3. Take the next 16 points at 22.5° yaw angles from the previous point.
4. Invert unit with Roll at 180° and take pitch to -45° and take point 7.
5. Take the next five points at 22.5° yaw angles from the previous point maintaining the 180° roll and the pitch at -45°.

A.1.2.2 Manual calibration with non-inverted points.
Sparton’s Manual calibration algorithm is best used when the points are as diversely spread as possible for a given environment. Here is an example without inverted points. The order of points taken does not matter.

1. Start by finding the maximum magnetometer X-Value in the X-Y plane and use this as the starting point
2. Take first point with Pitch increased to 45°.
3. Take the next 8 points at 45° yaw angles from the previous point.
4. Decrease Pitch to 22.5° and take point 9.
5. Take the next 8 points at 45° yaw angles from the previous point maintaining the 22.5° pitch.
6. Decrease Pitch to -22.5° and take point 17.
7. Take the next 8 points at 45° yaw angles from the previous point maintaining the -22.5° pitch.
8. Decrease Pitch to -45° and take point 25.
9. Take the next 8 points at 45° yaw angles from the previous point maintaining the -45° pitch.
B  How and When to use the In-Field Calibration related variables

B.1  magFieldCalErr
The magnetic data collected during field calibration will be analyzed for variability in field strength. This variability can be related directly to heading error providing instant feedback as to the quality of calibration (called magFieldCalErr). This variable gives an estimate of the error in degrees and is computed once at the end of the in-field calibration process.

- **Usage:**
  - Execute the Manual or 2D in-field calibration to completion
  - During the convergence
    - Monitor this variable to see if the calibration algorithm has converged.
  - After the user has stopped the convergence
    - Read the variable magFieldCalErr (for example: magFieldCalErr di.) If the value is too high, then perform the in-field calibration again with better point selection

B.2  yawErrEst
This is the estimated heading error from AdaptNavIII. It includes the sensor variances within AdaptNav along with the magFieldCalErr above.

- **Usage:**
  - This variable should be monitored during real-time operation.
  - Monitor this variable to determine if the magnetic/accelerometer/gyroscope based heading is off course. The output will be in possible degrees of heading error.

B.3  absoluteYawErr
This is the heading error between the output of AdaptNavIII and the current magnetic measurements coming from the magnetometers. Since AdaptNavIII will rely on the gyros during periods of magnetic distortion, the user may not know they are in a degraded environment by looking at just yawErrEst. This variable will tell them they are in a bad environment and need to move to a different location if possible.

- **Usage:**
  - This variable should be monitored during real-time operation.
  - Monitor this variable to determine if the total solution is off course. The output will be in possible degrees of heading error.