GEDC-6 Navigation Sensor -- Application Note AN1005

This Application Note details environmental noise characterizations used for drift compensation of heading, pitch and roll when in electrically and mechanically 'noisy' environments. Electrically 'Noisy' environments include magnetic materials in proximity of the navigation sensor, batteries, electric motors and high-current carrying wires. Mechanically noisy environments include vibration and vibratory rotation. Also includes select adjustments to AdaptNav™ sensor fusion parameters for improved overall performance.

Note: This Application Note (and the associated NorthTek™ script noted within) is not applicable to the Sparton DC-4 navigation sensor. The AHRS-8 navigation sensor has these environmental noise characterizations (real-time) built into the AdaptNav II™ sensor fusion algorithm.

Introduction

Magnetic navigation sensors measure the Earth's local magnetic field and acceleration due to gravity to determine an accurate heading. Any magnetic material that would interfere with or mask the intensity or direction of the Earth's magnetic field will affect navigation sensor performance. The following design guidelines should be considered when integrating the navigation sensor into an application platform:

- Install navigation sensor away from magnetic materials such as iron or steel. If the navigation sensor is to be enclosed, use aluminum, copper, or other non-magnetic material. Please note such an enclosure does not shield the navigation sensor from magnetic interference.
- Batteries are magnetic and should be kept as far away as possible.
- Permanent magnets and electric motors generate both static and time-varying magnetic disturbances and should be kept away from the navigation sensor
- High-current carrying wires can also generate static and time-varying magnetic fields and should be routed away from the navigation sensor

Sparton recognizes the challenges of navigation sensor integration into the application platform, and that these guidelines cannot always be met (or are met marginally) due to application platform size or other design constraints. These 'noisy' environments may require special compensation considerations.

This application note describes the adaptnavinit.4th NorthTek™ script product enhancement which allows users operating in 'noisy' environments to compensate for these effects by eliminating navigation sensor integration errors in the heading (yaw), pitch, and roll which appear to the user as a continuing drift.

User need for script

5612 Johnson Lake Rd • Deleon Springs, FL • 32130
800.824.0682 • 386.985.4631
www.sparton.com/ne • www.spartonnavex.com

Many environments have some sort of noise such as motors, vibration, or vibratory rotation. All users of the Sparton GEDC-6 navigation sensor can benefit from the use of this product enhancement, and Sparton strongly recommends its use to provide superior product performance.

Adaptnavinit.4th NorthTek™ Script

Absolute Deviation of Environmentally-Induced Noise

The absolute deviation of each set of sensors is measured. Once the measurements are complete (10 seconds), the script will perform gyro offset compensation (30 seconds). Do not move the navigation sensor at any time while the script is running. When the measurements are complete, the appropriate AdaptNav™ parameters are calculated and stored in non-volatile RAM for availability through power cycling. The content of the adaptnavinit.4th NorthTek™ product enhancement script is shown below.

The actual program script file is available for customer download at:
www.spartonnavex.com/technical-support/downloads/

```
// AdaptNav script for calculating sensor mean deviation

// AdaptNav Initialization
forget mx1

// Define Variables
// Magnetometers
variable mx1
variable mx2
variable mvx
variable my1
variable my2
variable mvy
variable mz1
variable mz2
variable mvz
variable mtot

// Accelerometers
variable ax1
variable ax2
variable avx
variable ay1
variable ay2
variable avy
variable az1
variable az2
variable avz
variable atot

// Gyros
variable gx1
variable gx2
variable gvx
```

```
variable gy1
variable gy2
variable gvy
variable gz1
variable gz2
variable gvz
variable gtot
variable filterK


// Restore Factory Defaults
: restoreDefaults
km0 f0.005 set drop
ka0 f0.005 set drop
wlim f0.0035 set drop
mlim f2.0 set drop
alim f40.0 set drop
emlim f0.1 set drop
ealim f0.1 set drop
kgyrooffset f1.0 set drop
;


: deviation

// Initialize variables
f0.04 filterK !
kgyrooffset f0.1 set
km0 f0.01 set drop
ka0 f0.01 set drop
wlim f0.002 set drop
emlim f0.1 set drop
ealim f0.1 set drop

    1000 0 do

    10 delay

// Process Magnetometers
       mx1 @
       magp &di @ dup
       mx2 @ f+ f2.0 f/ mx1 @ f- fabs
       mvx @ f- filterK @ f* mvx @ f+ mvx !
       mx1 ! mx2 !

       my1 @
       magp &di 4 + @ dup
       my2 @ f+ f2.0 f/ my1 @ f- fabs
       mvy @ f- filterK @ f* mvy @ f+ mvy !
       my1 ! my2 !

       mz1 @
       magp &di 8 + @ dup
       mz2 @ f+ f2.0 f/ mz1 @ f- fabs
       mvz @ f- filterK @ f* mvz @ f+ mvz !
       mz1 ! mz2 !

       mvx @ mvy @ mvz @ f+ f+ f3.0 f/ mtot !
```

```
// Process Accelerometers
       ax1 @
       accelp &di @ dup
       ax2 @ f+ f2.0 f/ ax1 @ f- fabs
       avx @ f- filterK @ f* avx @ f+ avx !
       ax1 ! ax2 !

       ay1 @
       accelp &di 4 + @ dup
       ay2 @ f+ f2.0 f/ ay1 @ f- fabs
       avy @ f- filterK @ f* avy @ f+ avy !
       ay1 ! ay2 !

       az1 @
       accelp &di 8 + @ dup
       az2 @ f+ f2.0 f/ az1 @ f- fabs
       avz @ f- filterK @ f* avz @ f+ avz !
       az1 ! az2 !

       avx @ avy @ avz @ f+ f+ f3.0 f/ atot !

// Process Gyros
       gx1 @
       gyrop &di @ dup
       gx2 @ f+ f2.0 f/ gx1 @ f- fabs
       gvx @ f- filterK @ f* gvx @ f+ gvx !
       gx1 ! gx2 !

       gy1 @
       gyrop &di 4 + @ dup
       gy2 @ f+ f2.0 f/ gy1 @ f- fabs
       gvy @ f- filterK @ f* gvy @ f+ gvy !
       gy1 ! gy2 !

       gz1 @
       gyrop &di 8 + @ dup
       gz2 @ f+ f2.0 f/ gz1 @ f- fabs
       gvz @ f- filterK @ f* gvz @ f+ gvz !
       gz1 ! gz2 !

       gvx @ gvy @ gvz @ f+ f+ f3.0 f/ gtot !

     loop

;

: init
cr ." Analyzing Navigation sensor Environment (10 Seconds) " cr
deviation
mlim mtot @ f6.0 f* set drop
alim atot @ f6.0 f* set drop
cr ." Computing Gyro Offsets, Do Not Move Unit (30 Seconds) " cr
InvokeGyroOffsetCal 1 set
30000 delay
cr ." Initialization Complete." cr
1000 delay
reset
```

```
;
```

Want to know more?

- Check it out here: www.spartonnavex.com