NXP Semiconductors
DATA SHEET: PRODUCT PREVIEW

Document Number: NSFK_DS
Rev. 0.9, 11/2016

NXP Sensor Fusion Library for Kinetis MCUs

Updated for NXP Sensor Fusion Release 7.10

Contents

1 Introduction

Sensor Fusion is the process where data from several different sensors are *fused* to complete computations that a single sensor could not handle. An example of sensor fusion is computing the orientation of a device in 3-dimensional space using an accelerometer and magnetometer. That data might then be used to alter the perspective presented by a 3D GUI or game.

The NXP Sensor Fusion Library for Kinetis MCUs provides advanced functions for computation of device orientation, linear acceleration, gyroscope offset and magnetic interference based upon the outputs of NXP inertial and magnetic sensors.

Features

- Supports:
  — Accelerometer only (roll, pitch and tilt)
  — Magnetometer only (2D auto)
  — Gyro only
  — Accelerometer plus magnetometer (eCompass)
  — Accelerometer plus gyro (gaming)
  — Accelerometer plus magnetometer plus gyroscope sensors
- Includes NXP's award-winning[1] magnetic compensation software
  — Provides geomagnetic field strength, hard- and soft-iron corrections, and quality-of-fit indication
- Very low power consumption
  — 1.4 mA 9-axis fusion I_{DD} on Kinetis ARM Cortex M4F devices at 40 Hz fusion rate
  — 0.4 mA 6-axis fusion I_{DD} on Kinetis ARM Cortex M4F devices at 40 Hz fusion rate
- Programmable sensor sample and fusion rates
- Supports multiple 3D frames of reference (aerospace NED, Android and Windows 8)
- Library is coded in standard C99 ANSI C
- Compatible with the NXP Sensor Fusion Toolbox for Android and Windows
- Included as part of the Kinetis Software Development Kit (KSDK)
- Out-of-the box support for the following Freedom development platforms mated with either FRDM-FXS-MULT2-B or FRDM-STBC-AGM01 sensor boards
  — FRDM-K64F
  — FRDM-K22F
- Library version 7.10 includes support for both bare-metal and RTOS-based projects.

Typical applications

- Notebook, tablet and smartphone sensor fusion
- Gaming, motion control, head-mounted displays, wearable electronics
- Air mouse, remote control
- Navigation, eCompass, IoT (Internet of Things) sensor data management

1 Freescale (now NXP) received the Electronic Products Magazine 2012 Product of the Year Award for our eCompass software.

Table 1. Feature comparison of the NXP Sensor Fusion Algorithm options

Feature	Accel only	Mag only	Gyro Only	Accel + gyro	Accel + mag	Accel + mag + gyro
Filter type	Low pass	Low pass	N/A	Indirect Kalman	Low pass1	Indirect Kalman
Roll / Pitch / Tilt in degrees	Yes	No	Yes	Yes	Yes	Yes
Yaw in degrees	No	Yes	Yes	No	Yes	Yes
Angular rate[2] in degrees/second	virtual [2] axis[3]	Yaw only	Yes	Yes	virtual [3] axis	Yes
Compass heading (magnetic north) in degrees	No	Yes	No	No	Yes	Yes
Quarternion and rotation vector	Yes	Yaw only	Yes[4]	Yes	Yes	Yes
Rotation matrix	Yes	Yaw only	Yes[4]	Yes	Yes	Yes
Linear acceleration separate from gravity	No	No	No	Yes	No	Yes
Aerospace (North-East-Down) frame of reference	Yes[5]	Yes	Yes[5]	Yes[5]	Yes	Yes
ENU (Windows® 8 variant) frame of reference	Yes[5]	Yes	Yes[5]	Yes[5]	Yes	Yes
ENU (Android variant) frame of reference	Yes[5]	Yes	Yes[5]	Yes[5]	Yes	Yes
Magnetic calibration included	N/A	Yes	N/A	N/A	Yes	Yes
Gyro offset calibration included	N/A	N/A	No	Yes	N/A	Yes
FRDM-K64F board support	Yes	Yes	Yes	Yes	Yes	Yes
FRDM-K22F board support	Yes	Yes	Yes	Yes	Yes	Yes

1. More precisely: a nonlinear modified exponential low pass quaternion SLERP filter.
2. Angular rate for 6 and 9-axis configurations with a gyro include corrections for gyro offset.
3. Subject to well-known limitation of being blind to rotation about axes aligned with gravity.
4. Rotations for gyro-only solution are relative to sensor position (there is no global frame).
5. These solutions do not include a magnetometer, therefore there is no sense of compass heading.

Table 2. Feature Options

Feature	Options
License	BSD 3-Clause
CPU selection[1]	MK64FN1M0VLL12 MK22FN512VLH12
Board customizable	Yes
Sensor sample rate	Programmable
Fusion rate	Programmable
Frame of reference	Programmable
Algorithms executing	Programmable
MCU sleep mode enabled between samples/calculations	No. It can be added by user, but will interfere with UART operation.
Power savings mode when stationary[2]	Supported
RTOS	FreeRTOS
Code flexibility	Full source code is provided. All files are can be modified.
Product deliverables	Delivered as part of the Kinetis Software Development Kit for selected platforms. Sensor Fusion specific deliverables include: • This datasheet • Software user guide • Extensive set of application notes • Preconfigured example applications • All source code

1. Listed MCUs are those supported by included project templates. The fusion library should be portable to any ARM® processor without change. Ports to other architectures are generally very straightforward, as the library is written in standard C.
2. Power savings when stationary is added at the application level. Example code is included in the sensor fusion user guide.

2 Functional Overview

2.1 Introduction

Sensor fusion encompasses a variety of techniques that:

- Trade off strengths and weaknesses of the various sensors to compute something more than can be calculated using the individual components
- Improve the quality and noise level of computed results by taking advantage of:
  — Known data redundancies between sensors
  — Knowledge of system transfer functions, dynamics and kinematics

The NXP Sensor Fusion Library for Kinetis (Fusion Library) supports several combinations of sensors. In general, performance improves as more sensors are added to the system. The primary function of the library is to compute orientation of a sensor subsystem relative to a global frame of reference.

Orientation can be expressed in a number of different ways:

- Tilt from vertical (may also be expressed as roll + pitch)
- Compass heading (geomagnetic north)

- Full 3D rotation from a global frame in any of the following forms:
  — Rotation matrix
  — Rotation vector (3D axis of rotation and rotation about that axis)
  — Quaternion
  — Euler angles (roll, pitch and yaw)

For additional portability details and guidelines, refer to the *NXP Sensor Fusion for Kinetis MCUs User Guide*, which is part of the *Sensor Fusion Library* installation. The Fusion Library is distributed in source code form and is designed to sit on top of board abstractions provided by the KSDK.

2.2 Accelerometer Only

An accelerometer measures linear acceleration minus gravity. If linear acceleration is zero, this sensor can be used to measure tilt from vertical, roll and pitch. Computation of yaw is not supported by this configuration.

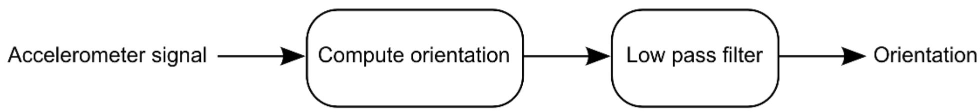


Figure 1. Accelerometer only block diagram

2.3 Accelerometer Plus Magnetometer

The accelerometer plus magnetometer configuration is often used as an electronic compass. The electronic compass is subject to the linear acceleration equals zero, assumption. Accuracy is dependent upon negligible magnetic interference from the environment in which the sensors travel.

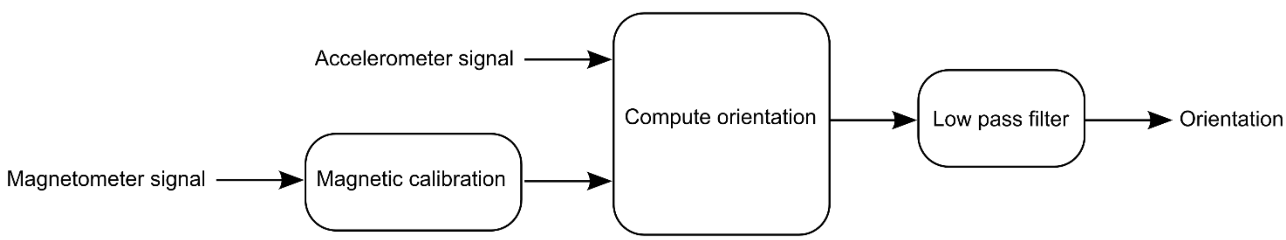


Figure 2. Accelerometer plus magnetometer block diagram

2.4 Accelerometer Plus Gyroscope

Using a gyroscope in addition to an accelerometer yields the ability to smoothly measure rotation in 3D space, although the system can only yield orientation to some random horizontal global frame of reference. That is, the system has no sense of magnetic north. Computation of yaw is not supported by this configuration.

This configuration is commonly known as an *Inertial Measurement Unit* (IMU).

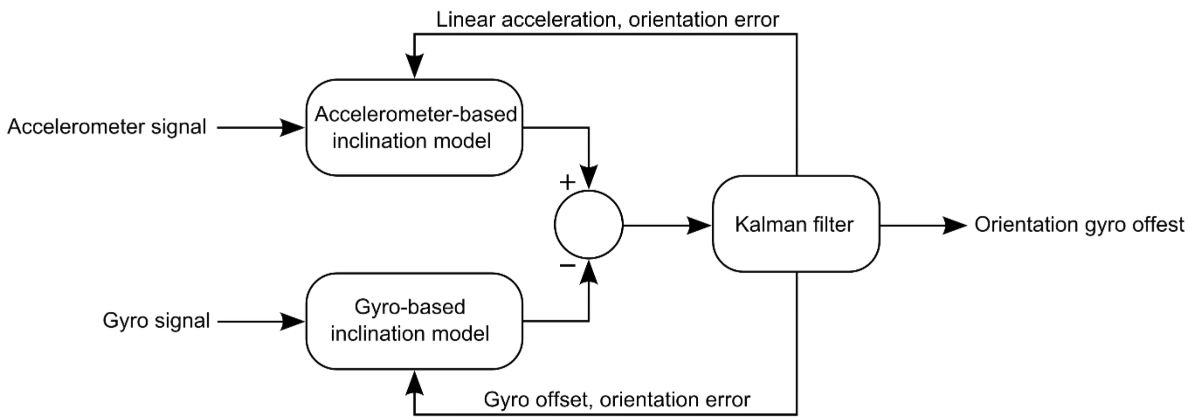


Figure 3. Accelerometer plus gyroscope block diagram

2.5 Accelerometer Plus Magnetometer Plus Gyroscope

Sometimes referred to as *Magnetic, Angular Rate and Gravity* (MARG), this subsystem offers an optimal combination of sensors for smooth tracking of orientation and separation of gravity and linear acceleration. This system is capable of yielding absolute orientation data with respect to magnetic north.

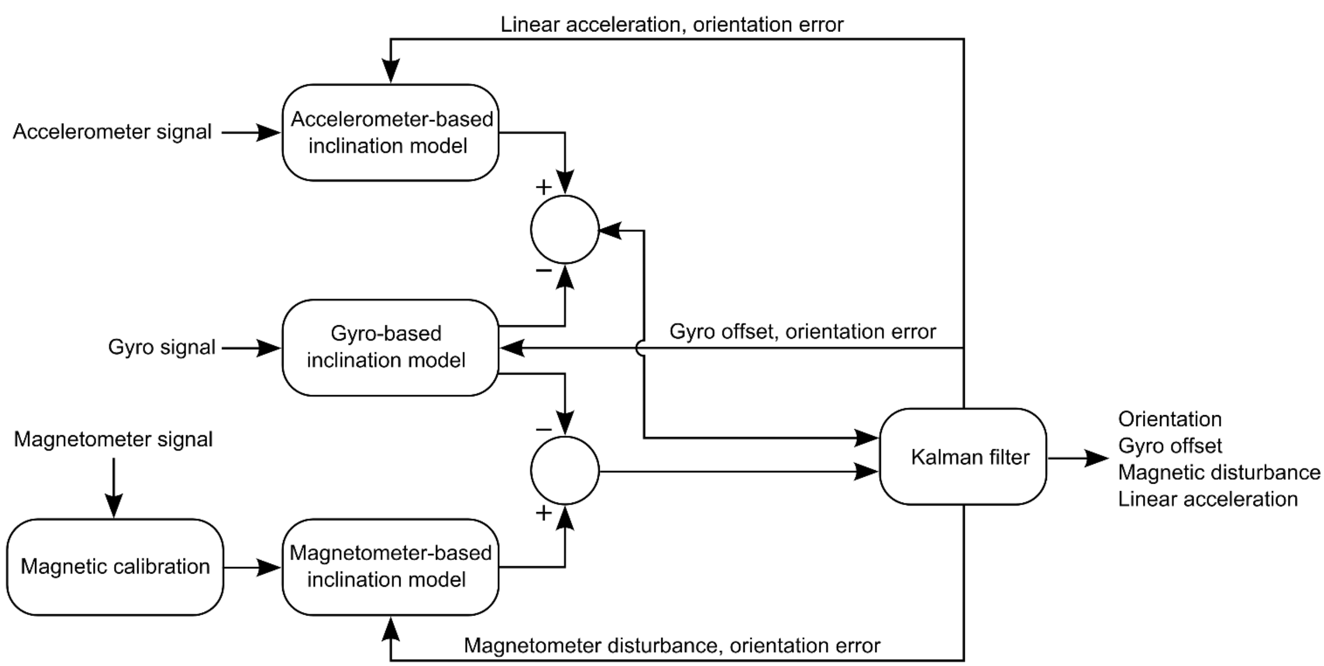


Figure 4. Accelerometer plus magnetometer plus gyroscope block diagram

3 Additional Support

NXP Sensor Fusion Toolbox provides support for both Android and Windows operating systems. See Table 3 for the differences between the two implementations.

Table 3. NXP Sensor Fusion Toolbox features by platform

Feature	Android	Windows
Bluetooth wireless link	✓	Requires BT on PC (built-in or dongle)
UART over USB	—	✓
OS requirements	>=Android 4.0.3	>=Windows 7.0
Support for native sensors	✓	—
Device View	✓	✓
Panorama View	✓	—
Statistics View	✓	—
Canvas View	✓	—
Orientation Value vs Time Plots	—	✓
Inertial Value vs Time Plots	—	✓
Magnetics	—	✓
Kalman Parameter Plots	—	✓
Altimeter Value vs Time Plots	—	✓
Data Logging	✓	✓
Integrated documentation	✓	✓
Availability	Google Play	NXP website
Price	Free	Free

3.1 NXP Sensor Fusion Toolbox for Android

The Fusion Library is supplied in the form of CodeWarrior projects for specific NXP development boards. The basic function of the Sensor Fusion Android implementation are shown in

Figure 5. These projects are compatible with the NXP Sensor Fusion Toolbox for Android, which can be freely downloaded from Google Play. For download and training links, visit nxp.com/sensorfusion.

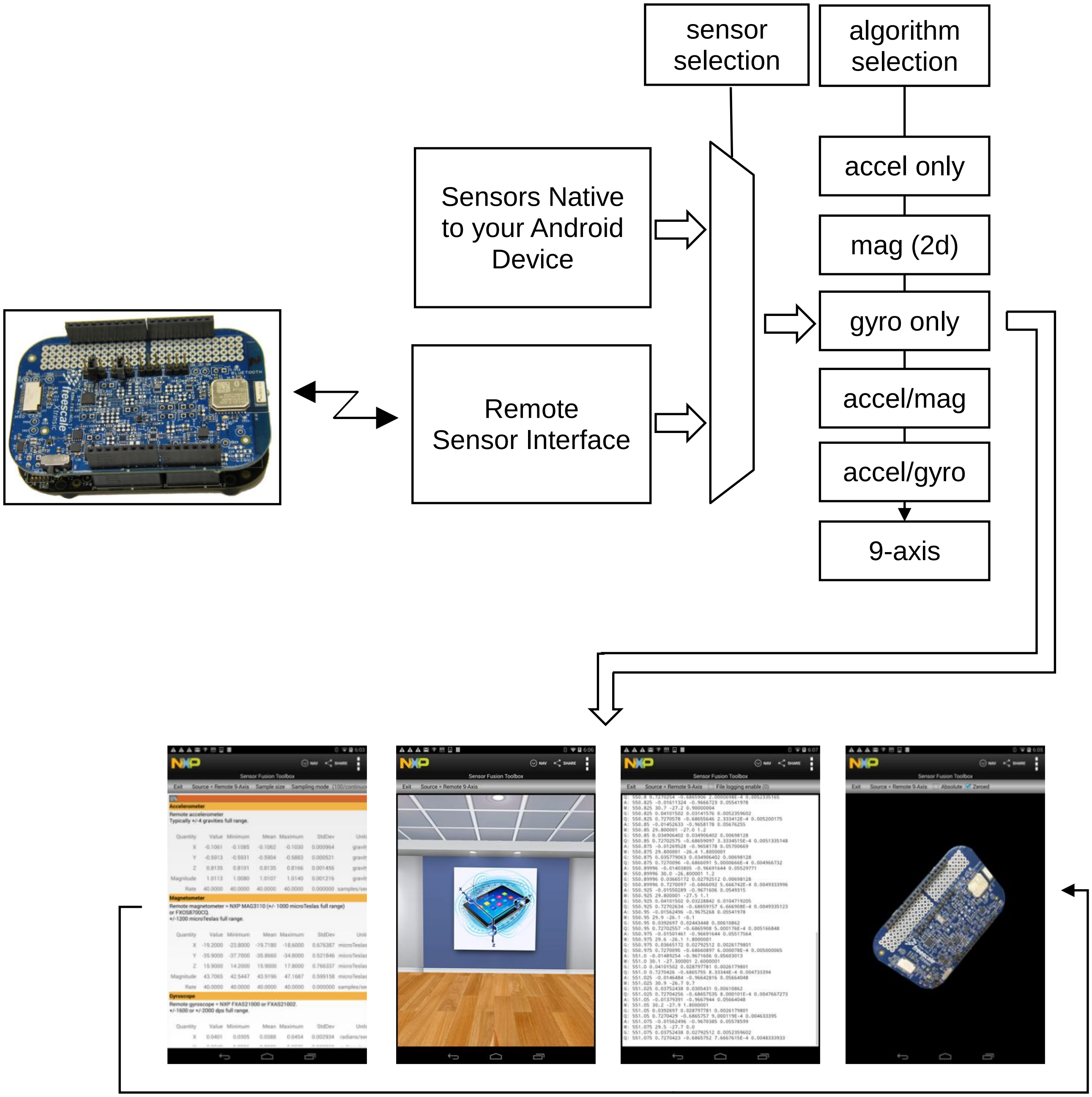


Figure 5. NXP Sensor Fusion Toolbox for Android basic functions

3.2 NXP Sensor Fusion Toolbox for Windows

The Sensor Fusion Toolbox includes an equivalent version of the software for Windows. For download and training links, visit nxp.com/sensorfusion.

***Note**: Be sure you have downloaded the NXP Sensor Fusion Toolbox for Windows build date 23 June 2016 or later.*

Additional Support

Figure 6 and Figure 7 are Sensor Fusion Toolbox screenshots of the Windows version[2].

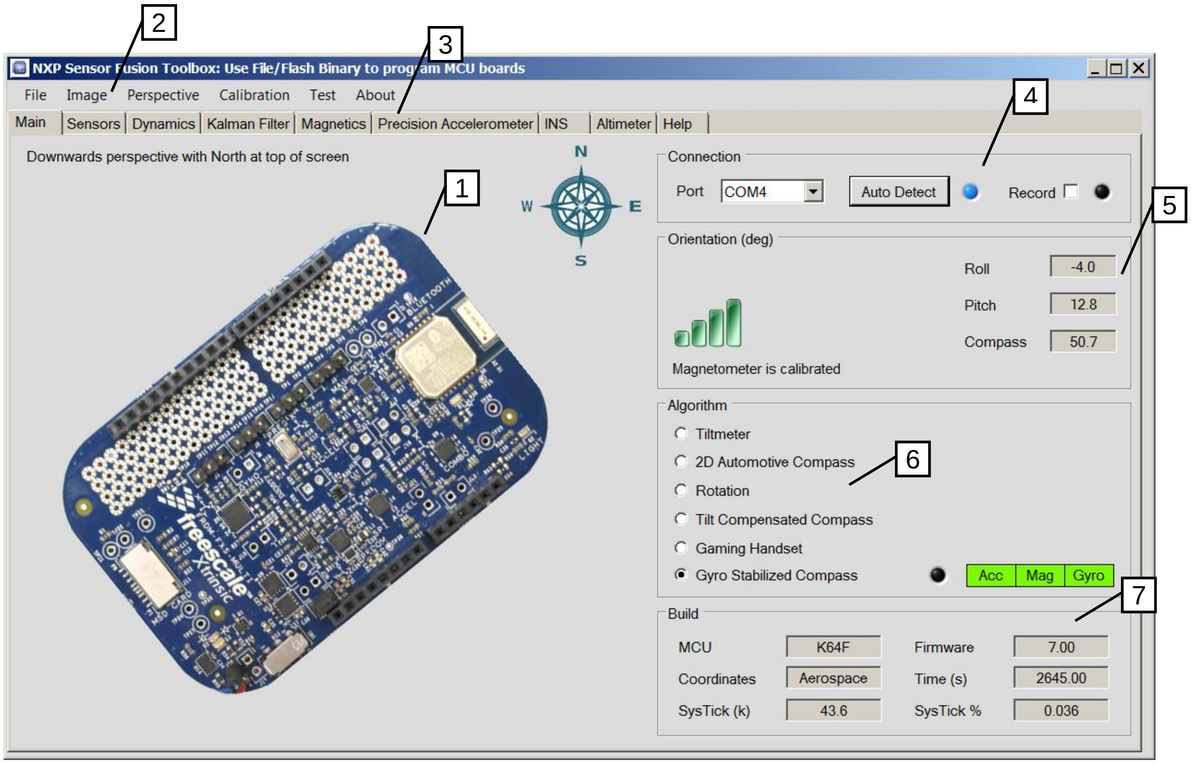


Figure 6. PC Version - Main Tab

Figure 6 is a snapshot of the Main Tab from the Windows version of the Sensor Fusion Toolbox. It has the following components:

1. Rotation 3D PCB display
2. Pull-down navigation menus: File, Image, Perspective, Calibration, Test and About
3. Navigation tabs for:
    - Main (board view)
    - Sensors
    - Dynamics
    - Magnetics
    - Precision Acclerometer
    - INS
    - Altimeter
    - Help
4. Communications

2 This is the 23 June 2016 version of the tool. Appearances will vary slightly from version to version.

- Communications port selection
- Auto-detect button
- Activity indicator
- Record control for creating data logs

5. Roll / Pitch / Compass and Magnetic Calibration Status
6. Algorithm selector
7. Sensor board run-time and build parameters

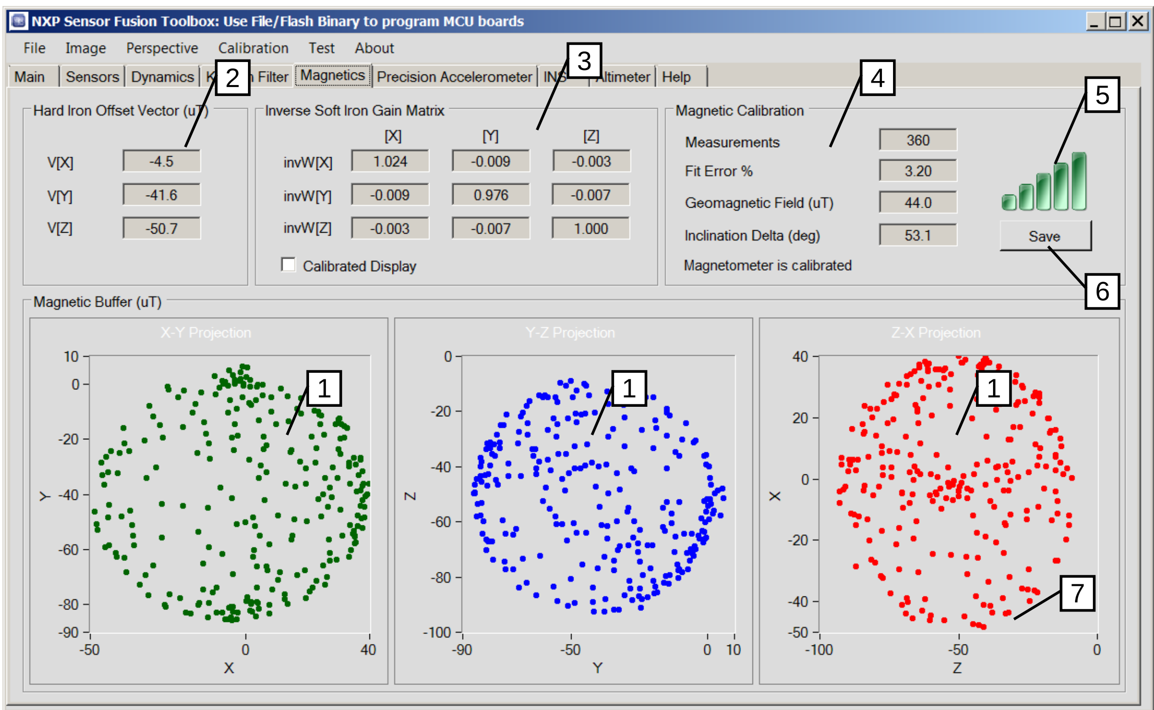


Figure 7. Windows version - Magnetics Tab

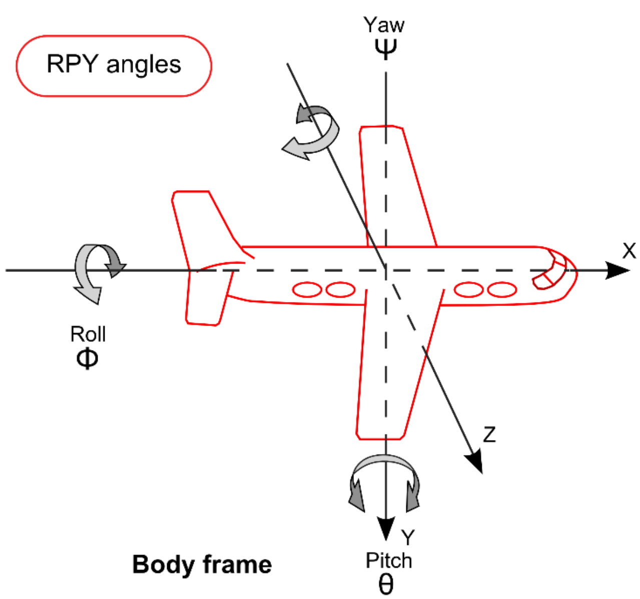
Figure 7 illustrates the Magnetics tab of the toolbox. Labelled features include:

1. 2D representation of the data point cloud used for hard/soft iron compensation
2. Computed hard iron vector
3. Soft iron matrix
4. Statistics
5. Calibration indicator
6. Save to text file control

3.3 Terms and Acronyms

Table 4: Terms and acronyms

Term	Definition

Term	Definition
DUT	Device Under Test
ENU	A global frame of reference described by X = **E**ast, Y = **N**orth, Z = **U**p
g	abbreviation for gravities. 1 standard gravity = 9.80665 m/s^2. Accelerometers are commonly trimmed using the local gravimetric field, which can vary by 1/2 percent depending upon altitude and latitude.
gauss	CGS system unit for measuring magnetic field strength. 100 μT = 1 gauss.
IMU	Inertial Measurement Unit = accelerometer + gyro
Kinetis	NXP family of ARM®-based MCUs
MagCal	Magnetic Calibration
MARG	Magnetic Angular Rate Gravity = IMU + magnetometer
MCU	Micro-Controller Unit
microTesla	The Tesla is the SI unit for measuring magnetic field strength. 1E–4 Tesla = 100 μT = 1 gauss
NED	A global frame of reference described by X = **N**orth, Y = **E**ast, Z = **D**own
pitch	In the Aerospace/NED frame of reference, defined as a rotation about the Y-axis
RHR	Right Hand Rule—a standard convention for describing the positive/negative sense of rotations about an axis of rotation. See http://en.wikipedia.org/wiki/Right_hand_rule.
roll	In the Aerospace/NED frame of reference, defined as a rotation about the X-axis
RPY	Roll, Pitch, and Yaw. In the Aerospace/NED frame of reference these are defined as rotations about the X-axis,Y-axis, and Z-axis 
SI	International System of Units (meter, kilogram, second, ...)
SLERP	**S**pherical **L**inear int**ERP**olation - See http://en.wikipedia.org/wiki/SLERP
SysTick	A feature of the ARM® processor; a clock timer which for the purposes of this discussion, 1 sysTick = 1 CPU clock cycle.
tilt	Angle from vertical
UART	Universal Asynchronous Receiver / Transmitter, also known as SCI (Serial Communications Interface)
yaw	In the Aerospace/NED frame of reference, defined as a rotation about the Z-axis

3.4 References

Included in with the sensor fusion source code, you will find:

- This data sheet
- NXP Sensor Fusion for Kinetis MCUs User Guide
- NXP Application Note AN5016, Rev. 2.0: Trigonometry Approximations
- NXP Application Note AN5017, Rev 2.0: Aerospace, Android and Windows 8 Coordinate Systems
- NXP Application Note AN5018, Rev. 2.0: Basic Kalman Filter Theory
- NXP Application Note AN5019, Rev. 2.0: Magnetic Calibration Algorithms
- NXP Application Note AN5020, Rev. 2.0: Determining Matrix Eigenvalues and Eigenvectors by Jacobi Algorithm
- NXP Application Note AN5021, Rev. 2.0: Calculation of Orientation Matrices from Sensor Data
- NXP Application Note AN5022, Rev. 2.0: Quaternion Algebra and Rotations
- NXP Application Note AN5023, Rev. 2.0: Sensor Fusion Kalman Filters
- NXP Application Note AN5286, Rev. 2.0: Precision Accelerometer Calibrations

External references you may find useful include:

- nxp.com/sensorfusion

MCU on Eclipse blog at mcuoneclipse.com

- Kinetis Design Studio software at nxp.com/kds
- Euler Angles at en.wikipedia.org/wiki/Euler_Angles
- Introduction to Random Signals and Applied Kalman Filtering, 3rd edition, by Robert Grover Brown and Patrick Y.C. Hwang, John Wiley & Sons, 1997
- Quaternions and Rotation Sequences, Jack B. Kuipers, Princeton University Press, 1999
- NXP Freedom development platform home page at nxp.com/freedom
- OpenSDA User's Guide, NXP Semiconductors N.V., Rev 0.93, 2012-09-18
- NXP OpenSDA support page at nxp.com/opensda
- PE micro OpenSDA support page at pemicro.com/opensda
- Segger OpenSDA support page at segger.com/opensda.html
- Matlab-based simulation of object trajectories at https://github.com/memsindustrygroup/TSim

4 Test Environments

4.1 General Considerations

Fusion algorithms can be *tuned* to trade off one performance parameter versus another. Examples include:

- Speedy handling of magnetic interference versus slower convergence to magnetic north
- Smoothness versus responsiveness
- Accuracy versus bandwidth

All of the above means that there is no *one correct* configuration. Accordingly, this datasheet presents typical performance as observed on the sample projects supplied by NXP on specific NXP development platforms. No attempt has been made to measure a statistically significant number of boards. Measured values are typically those observed on as few as one board.

The six- and nine-axis Kalman filters are optimized for calculation of device orientation. Unmodelled sources of error will affect the other sensor outputs. Tradeoffs are a function of the Kalman filter configuration and are subject to change.

Note: *Performance metrics are simulated based on Version 7.xx of the sensor fusion library, and using basic sensor models, which include noise effects, but ignore nonlinearity and other factors. See* **Simulation Environment"** *for details.*

4.2 Limitations Imposed via Sensor Choice/Configuration

The table below specifies the sensor configuration used to determine parameters specified in this datasheet.

Table 5. Sensor/configuration imposed limitations

Characteristic	Symbol	Min	Max	Unit
Maximum angular rate	AR_{max}	–2000	+2000	dps
Linear acceleration	LA_{max}	–4	+4	*g*
Magnetic field	B_{max}	–1200	+1200	µT

It is believed that the fusion algorithms themselves have no upper limits on any of the parameters above. These are strictly limits associated with the specific physical sensors used to characterize the library.

4.3 Repeatability

Computation metrix and IDD measurements are generally very repeatable.

The 6-axis Kalman filter has no sense of absolute North (it does not use a magnetometer). This means it may include arbitrary rotations about the vertical axis. This could have the result of mixing global frame X & Y signals.

See "Orientation Error Sensitivity to Linear Acceleration" for an example of how algorithm response can be a function of the trajectory type.

4.4 Hardware Platforms

In the following subsections, some parametrics are measured, some represent simulated results. Hardware platforms used for benchmarking purposes are briefly outlined in the following subsections. All boards shown are described in more detail, and can be ordered at nxp.com/freedom.

4.4.1 Sensor Shields

The FRDM-MULT2-B sensor shield is designed to be mated to a variety of NXP Freedom development boards. It includes a number of sensor types, as well as a third-party Bluetooth communications module.

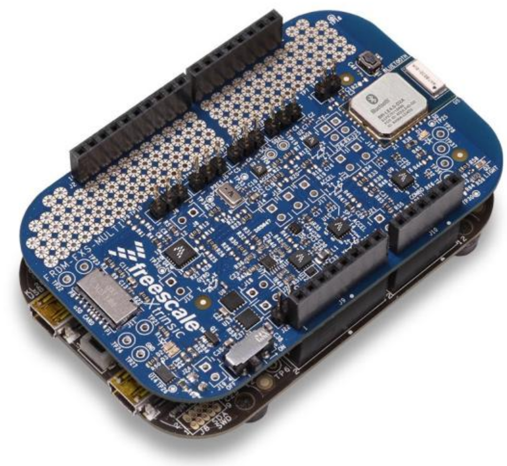


Figure 8. FRDM-KL25Z / FRDM-FXS-MULTI-B sensor fusion prototype platform

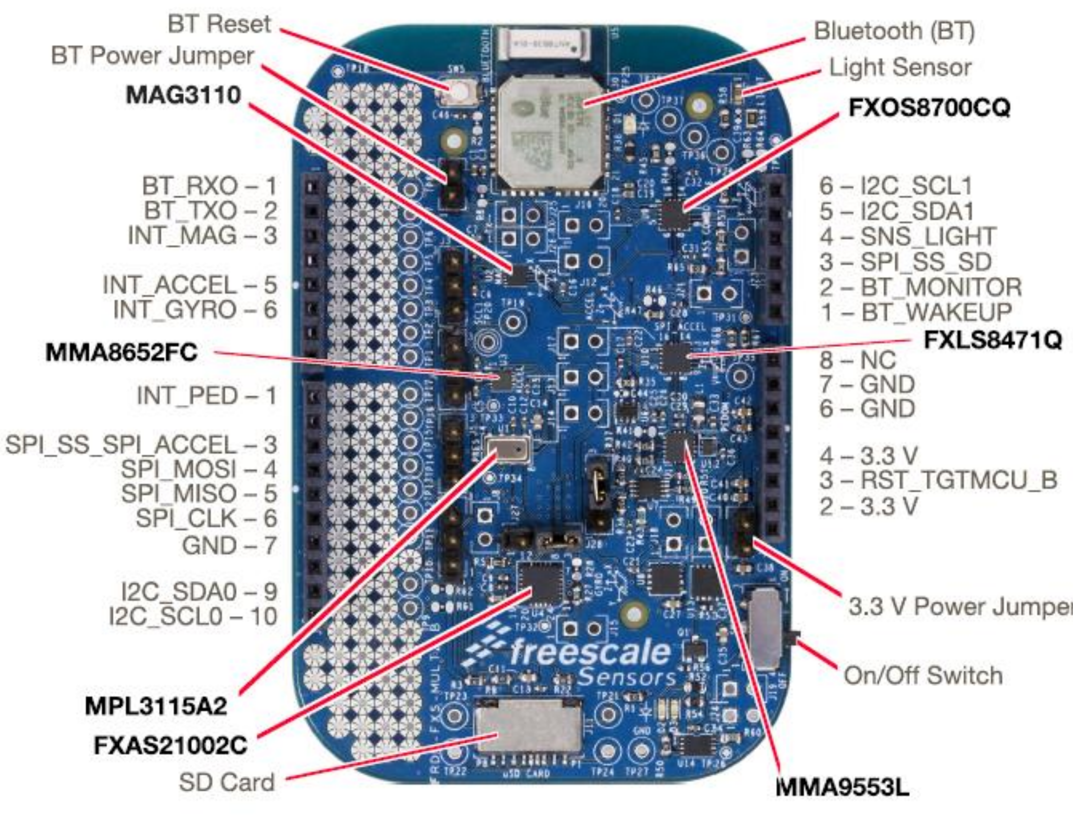


Figure 9. Expanded diagram of FRDM-FXS-MULT2-B sensor board

The FRDM-STBC-AGM01 can be viewed as a proper subset of the FRDM-MULT2-B. FRDM-AGM01 Sensors are also present on the FRDM-MULT2-B. The FRDM-AGM01 does not include wireless communications capabilities.

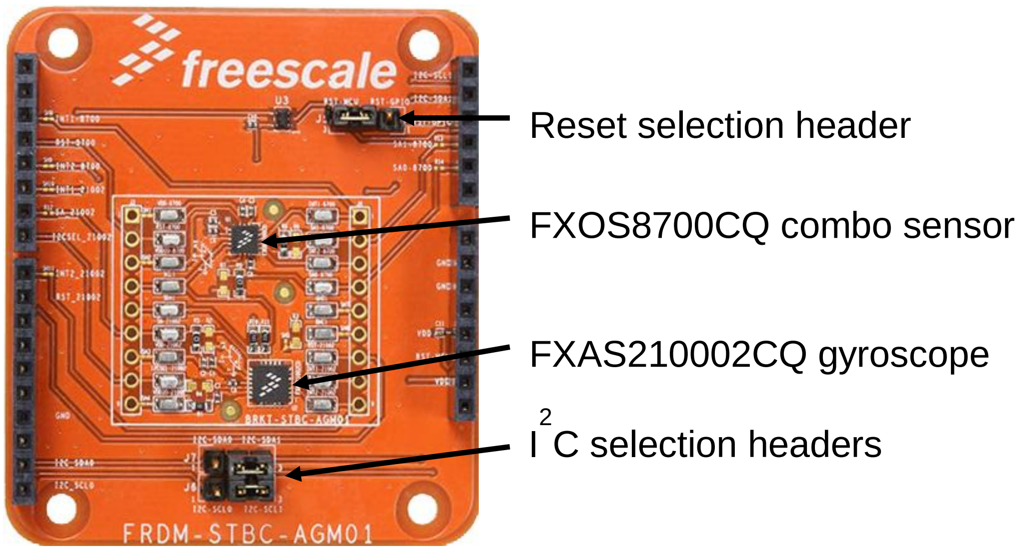


Figure 10: FRDM-STBC-AGM01

4.4.2 **Freedom Development Platforms**

The initial release of Version 7.00 of the Sensor Fusion library supports FRDM-K64F and FRDM-K22F Freedom development platforms. The two boards are similar in capabilities, although the K64F includes an Ethernet port, whereas the K22F does not.

Shields and base boards can be mixed and matched.

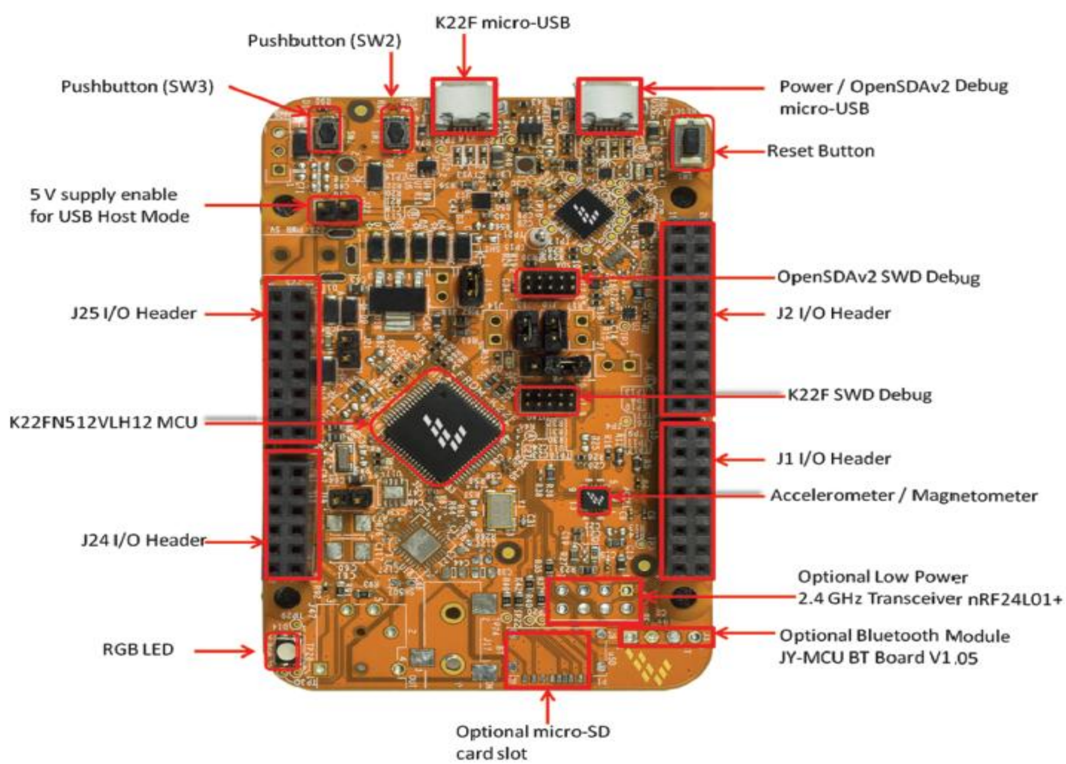


Figure 11: FRDM-K22F

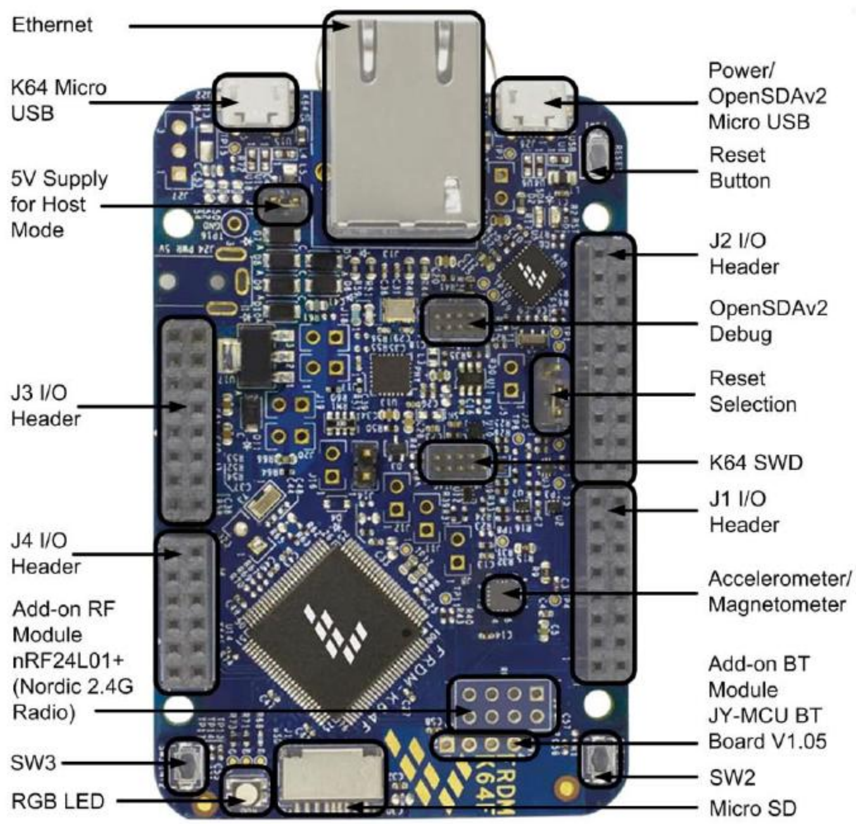


Figure 12: FRDM-K64F

4.4.3 'Standard' Board Orientations

If you think of the board stackup as a 6 sided box, then there are six different orientations that can be reached via simple one-axis rotations. The figure below utilizes ENU terminology, but the concept obviously applies to all frames of reference. Test sequences in this document are preconditioned by putting the DUT into one of these orientations (usually zero rotation to the global frame). Some sequences are repeated for every one of the six.

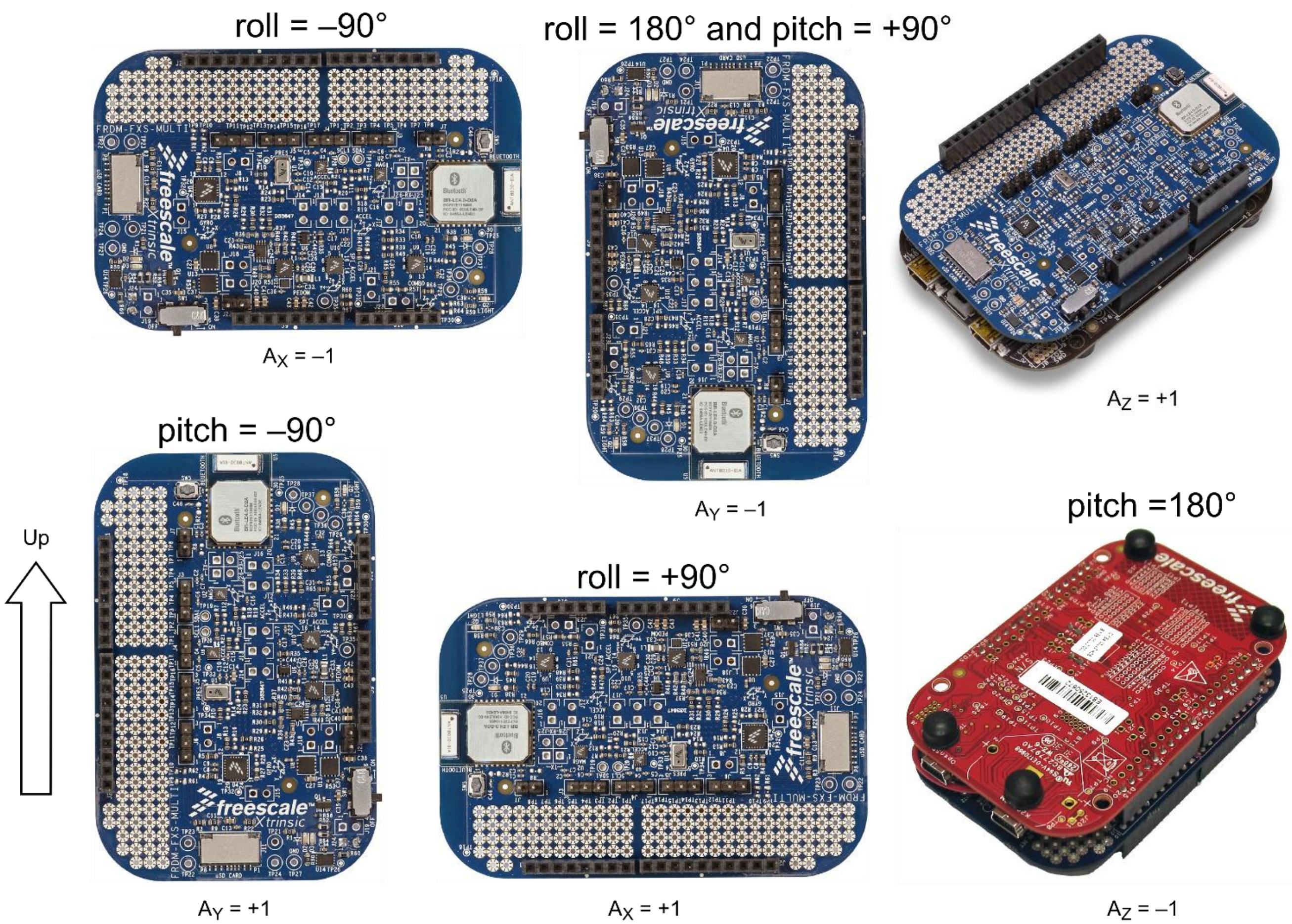


Figure 10. Major orientations relative to gravity

4.5 Simulation Environment

Many parameters are difficult, if not impossible, to reliably measure in a lab or production environment. Ambient magnetic fields vary tremendously in indoor environments. Determining filter sensitivities to various input parameters can only be done in a simulated environment.

Physical performance metrics presented in subsequent sections were simulated using the flow shown in the figure below:

1. Matlab is used to generate synthesized sensor values based upon predefined trajectory paths and environmental and sensor model parameters.

2. A version of the V7.xx sensor fusion has been modified to read the synthetic sensor values created above and then to write fusion results into a set of output files.
3. Matlab is then used to compare V7.xx synthesized orientation, magnetic calibration parameters, angular rates, etc. with the “known truth” values output by the first step above. The resulting reports were then used as source for the datasheet parameters that follow.

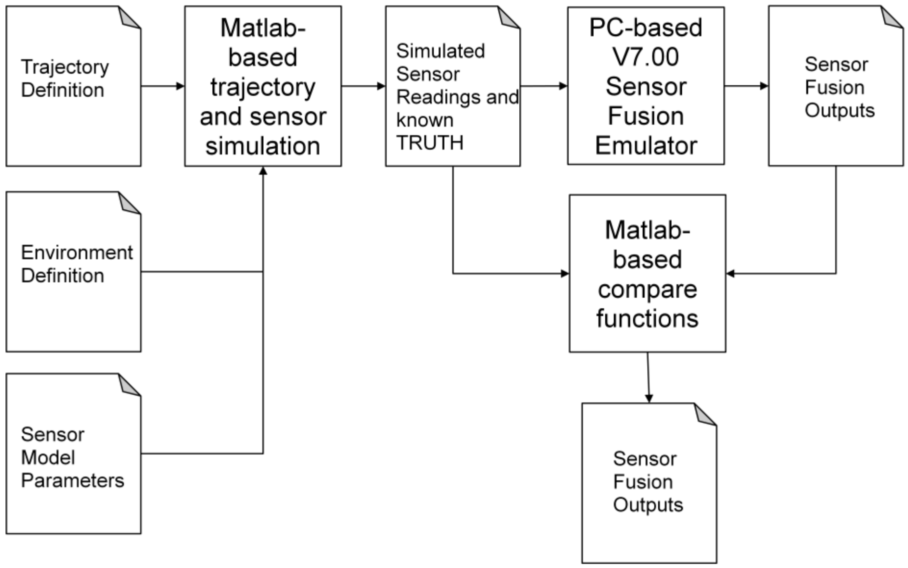


Figure 14: Sensor Fusion Simulation Environment

Unless stated otherwise elsewhere, parameters used to create the simulated environment are:

- Earth magnetic field corresponding to U.S. zip code 85284 (Tempe, Arizona) on 7 November 2013 as determined using the NOAA calculator at ngdc.noaa.gov/geomag-web
  — Declination: 10 degrees 39’ 36”
  — Inclination: 59 degrees 32’ 53”
  — Horizontal Intensity: 24.2976 μT
  — North Component (+N | -S): 23.8782 μT
  — East Component (+E | -W): 4.4945 μT
  — Vertical Component (+D | -U): 41.3285 μT
  — Total field: 47.9418 μT
- gravity = 9.80665 meters/second^2 = 1g
- constant temperature = 25C

Unless stated otherwise elsewhere, parameters used to create the simulated sensors are:

- Physical sensor dynamics are not modeled. Nor are latency effects associated with reading sensors via serial lines.
- Zero rotation orientation for all 3-axis sensors are assumed to be naturally aligned with the NED (+X=North, +Y=East, +Z=Down) frame of reference
- Ideal accelerometer model + simple noise + offset
  — Sample rate = 200 Hz

— 1 mg standard deviation offset on X/Y/Z. This is justified by assuming that the precision accelerometer calibration function has been utilized to remove post-board-mount offset.

— 100 μg/√Hz Gaussian noise on X/Y/Z

— +/- 4g range

— 14 bits ADC resolution

- Ideal magnetometer model + simple nose

— Sample rate = 200 Hz

— +/- 1200 μT range

— 16 bits ADC resolution

— X/Y-axis 1σ noise = 0.85 μT; Z-axis 1σ noise = 1.3 μT

— No hard/soft iron distortion

- Ideal gyroscope model + simple noise + offset

— Sample rate 200 Hz

— +/- 2000 dps range

— 16 bit ADC resolution

— 25 mdps/μg/√Hz Gaussian noise on X/Y/Z

— 15 LSB standard deviation offset on X/Y/Z

For all tests, the NXP Matlab-based Trajectory & Sensor Simulation Toolkit (TSim) is used to create DUT trajectories and simulated sensors readings.

Most tests require that a magnetic calibration procedure is run before the test to initialize the sensor fusion software magnetic buffer. Magnetic calibration is implemented as a trajectory made up of a sequence of random rotations lasting 30 seconds.

4.6 Frames of Reference

The table below summarizes the differences between the standard frames of reference supported by the fusion library. Simulations for this datasheet were run using the NED frame of reference.

Table 6. Frame of Reference Variations

	NED	Android	Windows 8
Axes alignment	NED	ENU	ENU
Angle rotation order	Yaw then pitch then roll	Yaw then roll then pitch	Yaw then pitch then roll
Gimbal lock	Roll instability (X-axis) at ±90 deg pitch (Y-axis)	Pitch instability (X-axis) at ±90 deg roll (Y-axis)	Roll instability (Y-axis) at ±90 deg pitch (X-axis)
Roll range	Clockwise[1] –180 to 180 deg	Anti-clockwise –90 to 90 deg	Clockwise[1] –90 to 90 deg
Pitch range	–90 to 90 deg	–180 to 180 deg	–180 to 180 deg
Yaw range	0 to 360	0 to 360	0 to 360
Compass heading	Yaw	Yaw	–Yaw

1. A clockwise rotation is defined as one that is positive in the Right-Hand-Rule (RHR) sense.

5 Specifications

All specifications and results in this document represent “typical” performance.

5.1 Power Specifications

5.1.1 Test Intent

MCU current is the average current consumption of the MCU executing the core fusion routines. This is obviously specific to the particular MCUs listed. This metric must be associated with a specific hardware configuration similar to those defined earlier in this document. Freedom Development Platform products are powered via the OpenSDA USB port for the results specified.

5.1.2 Procedure

This procedure uses modified versions of the standard demo build:

1. Measure (using the Sensor Fusion Toolbox) $sysTick_{fusion}$, which only includes time spent in the *core fusion routines*. It does *not* include:
    - Calls to magnetic calibration
    - Reading sensor data
    - Applying hardware abstraction layer
    - RTOS
    - Communications overhead
2. The relative ratio of time spent in the core fusion routines is

    $$fusion\_rate \times sysTick_{fusion} / \text{MCU clock rate}$$
3. I_{DD} current into the MCU is measured via power jumper on the board using a simple DVM[3]
4. Fusion I_{DD} = MCU I_{DD} x ratio computed above
5. Sample size = 1 board

For devices with floating point units, this may yield a somewhat optimistic number, as the computation makes the assumption that all MCU instructions consume the same amount of power. In fact, we can expect floating point instructions to consume a bit more.

5.1.3 Test Configuration

Table 7. I_{DD} / SysTick test case configurations

Component / Parameter	FRDM-K22F	FRDM-K64F
MCU	MK22FN512VLH12	Kinetis MK64FN1M0VLL12
CPU	ARM® Cortex M4 with Floating Point Unit	
CPU clock used for benchmarking	80 MHz	120 MHz
Bus clock used for benchmarking	40 MHz	60 MHz
RTOS	None – Bare Metal Implementation used to measure parametrics	
Compiler	IAR, Optimization Level = High (Balanced)	
Accelerometer	FXOS8700CQ	
Magnetometer		

[3] Consult the user guide / schematic for your Freedom board to determine the specific jumper number applicable to that board.

Specifications

Component / Parameter	FRDM-K22F	FRDM-K64F
Gyroscope	FXAS21000	
Accelerometer Sampling Rate	200 Hz	
Magnetometer Sampling Rate	200 Hz	
Gyroscope Sampling Rate	400 Hz	
Fusion Rate	40 Hz	
Magnetic Calibration	Serialized to run as part of the fusion loop	

5.1.4 **Results**

The parameters in tables that follow are measured using the configurations defined in Table 6 above. Numbers that follow do not include power consumed by sensor sampling or signal conditioning. They reflect only the core fusion algorithm itself.

Tables 8 and 9 were developed using Version 7.00 of the Sensor Fusion Library.

Table 8. Typical I_{DD} Executing on FRDM-K22F

Function	Fusion I_{DD} @ 40 Hz rate (mA)	Fusion I_{DD} / Hz (μA)
Accel only	0.08	2
2D Mag	0.07	1.8
Gyro only	0.05	1.4
Accel + Mag, eCompass	0.09	2.3
Accel + Gyro	0.41	10.3
9-axis	1.2	30.1

Table 9. Typical I_{DD} Executing on FRDM-K64F

Function	Fusion I_{DD} @ 40 Hz rate (mA)	Fusion I_{DD} / Hz (μA)
Accel only	0.13	3.3
2D Mag	0.09	2.1
Gyro only	0.06	1.6
Accel + Mag, eCompass	0.12	2.9
Accel + Gyro	0.42	10.5
9-axis	1.41	35.3

The currents in Table 10 were measured using a FreeRTOS-based project with sampling rates similar to the above, but with the addition of separate 40 Hz fusion and 200 Hz sensor sampling task to the RTOS-based project. The project is designed to shut down the gyro when no motion is detected by the accelerometer. Measured current includes IDD for FXOS8700CQ, FXAS21002 and dual 4.7K I^2C pullup resistors on the board.

Table 10. Typical Current Draw on AGM01 Sensor Board

Function	Measured $I_{DD\ P3V3}$ (mA)
Run Mode	3.4
Gyro Standby Mode	0.7

5.2 Computation Metrics

The Sensor Fusion Version 7.xx Library computations are measured directly using the Sensor Fusion Toolbox and the ARM® sysTick clock, using the configuration outlined in the previous section.

5.2.1 Clock Cycles

NXP used the built-in sysTick counter to measure each iteration of the fusion algorithms in units of CPU clock cycles.

The customer can repeat the exact same measurement, because both PC and Android Sensor Fusion Toolboxes display this information in a real-time basis. The test results may vary, depending on device movement, as presented numbers are typical.

Table 11. K SysTick values for Freedom development platforms and sensor combinations[1]

Freedom Platform	Accel	2D Mag	Gyro	eCompass	Accel +Gyro	9-axis
FRDM-K22F	3	2.7	2.1	3.6	16	46.7
FRDM-K64F	4	2.6	1.9	3.5	12.7	42.8

1. Based on “All Motions” build, which includes tilt, 2D automotive compass, rotation, eCompass, 6- and 9-axis Kalman filters. Does not include pressure/altimeter.

5.2.2 Memory Requirements

5.2.3 Intent

These parameters are total RAM and flash memory required to implement and execute the Fusion Library in a bare metal project.

This includes space for code storage, static and dynamic (stack) variables. This metric must be associated with a specific hardware configuration similar to those defined earlier in this document. It is also a function of the optimization level selected in the compiler.

5.2.4 Procedure

The projects for the binaries were created using IAR. Values for readonly code and data memory (flash) and readwrite data memory (RAM) were read directly from the c.map file generated by the toolset.

5.2.5 Results

Table 12. Memory (in bytes) requirements for "All Motion Algorithms" builds[1]

Freedom Platform	Read only code memory	Read only data memory	Read/write data memory[2]
FRDM-K22F	58,714	322	12,049
FRDM-K64F	58,430	322	12,057

2. Includes tilt, 2D automotive compass, rotation, eCompass, 6- and 9-axis Kalman filters. Does not include pressure/altimeter.
3. If your Read/write data memory numbers differ significantly from those shown above, it may be due to your choice of linker configuration file. The sensor fusion team has noted that the default files shipped with IAR give a much larger number than the IAR linker configuration files shipped with the NXP KDSK example projects.

Table 13. Memory (in bytes) requirements for FRDM-K22F and single algorithm

Algorithm	Read only code memory	Read only data memory	Read/write data memory
Accel (Tilt)	36,110	286	4,809
2D Mag	40,918	286	8,573
Gyro Only	28,886	286	3,553
Accel/Mag	47,882	286	10,005
Accel/Gyro	40,430	294	5,361
9-axis	52,854	294	11,101

5.3 Magnetic Calibration

5.3.1 Background

Hard-iron effects are due to magnetic materials in the vicinity of the sensor. These materials result in an apparent offset to sensor readings when the source of interference is fixed spatially, relative to the sensor.

For a given point in space, plotting magnetometer measurements at various sensor rotations results in the sphere shown on the right side of Figure 14. This makes sense, as the magnitude of the 3D magnetic field should not change just because the sensor is rotated.

Soft-iron effects result from the interaction of ferrous materials near the sensor interacting with the ambient magnetic field. If the source of soft-iron interference is again fixed spatially with respect to the sensor and does not demonstrate magnetic hysteresis, then the *sphere of plotted measurements* is distorted into an ellipsoid. This is shown, along with a hard-iron offset, on the left side of Figure 12.

The sensor fusion user guide provides background on the topic of hard- and soft-iron magnetic compensation. For the case where the sensor and sources of interference are spatially fixed with respect to one another, the distortions are linear, and can be reversed mathematically. This is the function of the NXP magnetic calibration library.

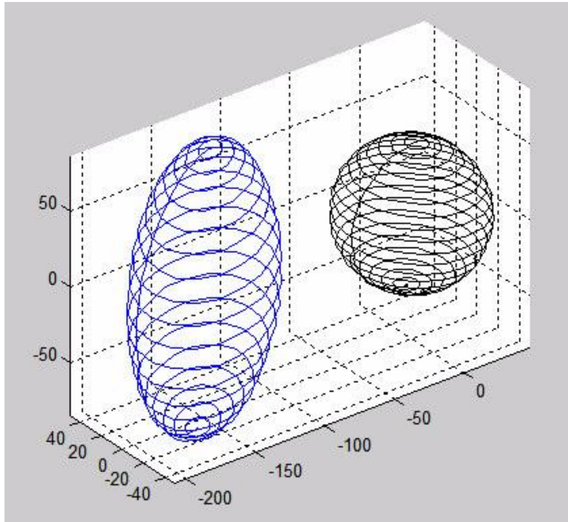


Figure 11. Distorted (left) and Corrected (right) Magnetic Field Data (simulated)

Both hard- and soft-iron interferences are a function of the sensor environment, and not the sensor itself. Each product design will inevitably result in different distortions.

Engineers assigned the task of physically designing PCBs and housings should pay careful attention to sources of magnetic interference early in the design phase.

Inductive charging films found in some portable devices exhibit a significant amount of magnetic hysteresis. This is a nonlinear phenomena and cannot be fully corrected by the NXP magnetic calibration library.

5.3.2 **The Magnetic Buffer**

The NXP sensor fusion utilizes three different versions of the calibration functions during startup. These are:

1. Compute hard iron offset only. Assume no soft iron offset. This is the “4-element model”.
2. Compute hard iron offset and diagonal terms of the soft iron matrix. This is the “7-element model”.
3. Compute hard iron and full soft iron matrix, including off-diagonal terms. This is the “10-element model”. It performs a total least squares fit of a number of data points to map the measured ellipsoid of measurements back into the ideal sphere.

Quality of fit improves as number and spacing of samples across the ellipsoid surface increases. There is a tradeoff in terms of data set size used for calibration versus CPU resources versus quality of fit. The fusion library includes heuristics which switch from 4 to 7 to 10 element models as additional data becomes available. Figure 16 below shows this progression for a nominal (no distortion) magnetic field over time. Figure 17 shows the progression for the case where we have 3X soft iron distortion in the Z direction. These curves will change as function of magnetic distortion and orientation over time.

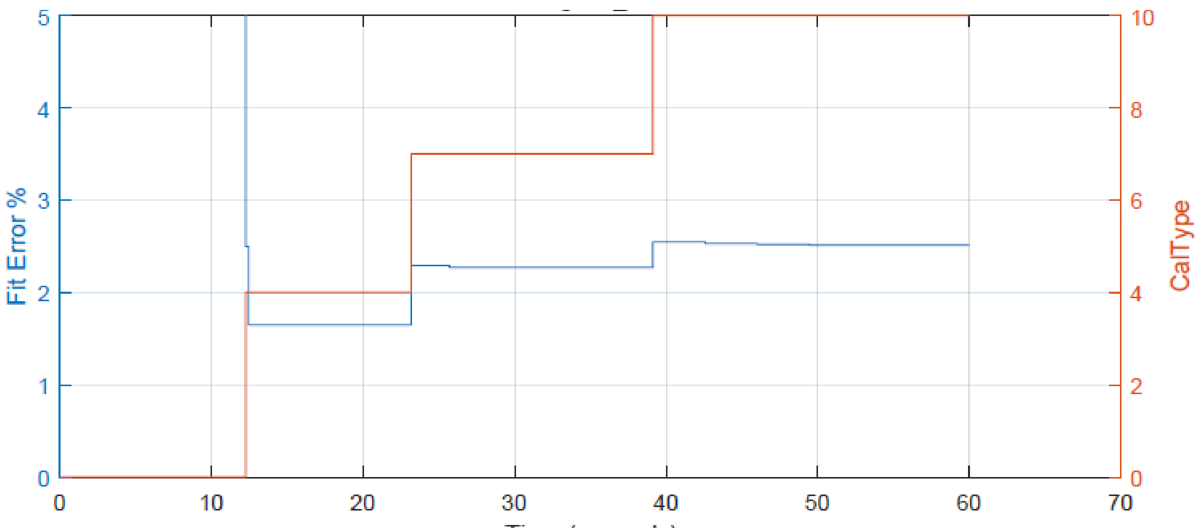


Figure 16: Fit error and calibration choice versus time (nominal case)

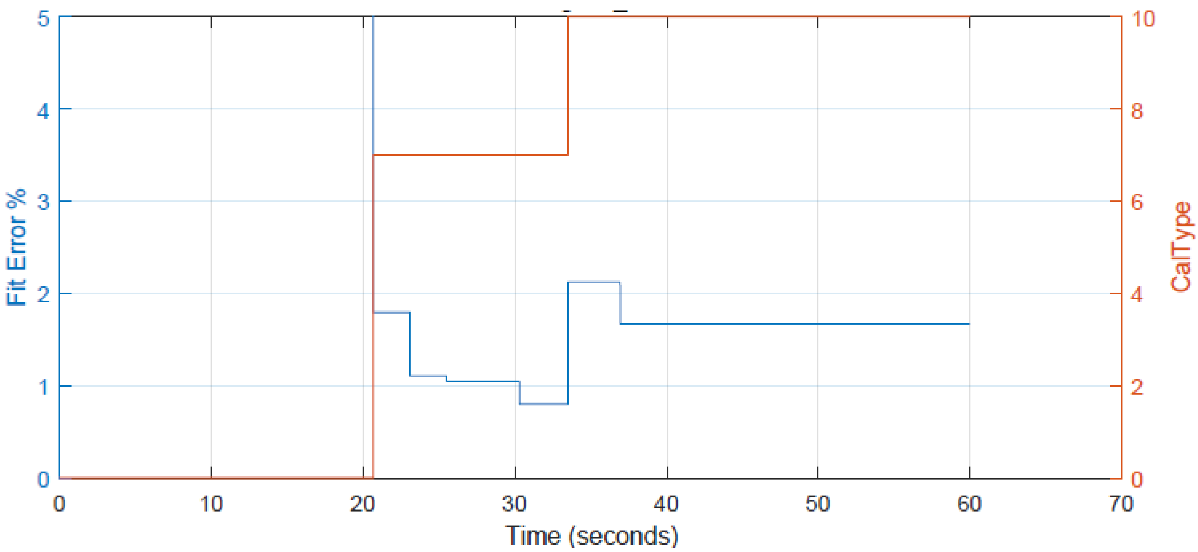


Figure 17: Fit error and calibration choice versus time (1:1:3 soft iron distortion)

5.4 Compass Heading Accuracy

5.4.1 Test Intent

Linearity is defined as the deviation of measured data from a least squares straight line approximation of that data.

Absolute accuracy is defined as the maximum difference between measured and ideal values.

These two concepts are illustrated in Figure 18.

Any non-linearities in compass heading curves map directly to non-linearities in the input sensors. The algorithms themselves will not introduced any steady state non-linearities.

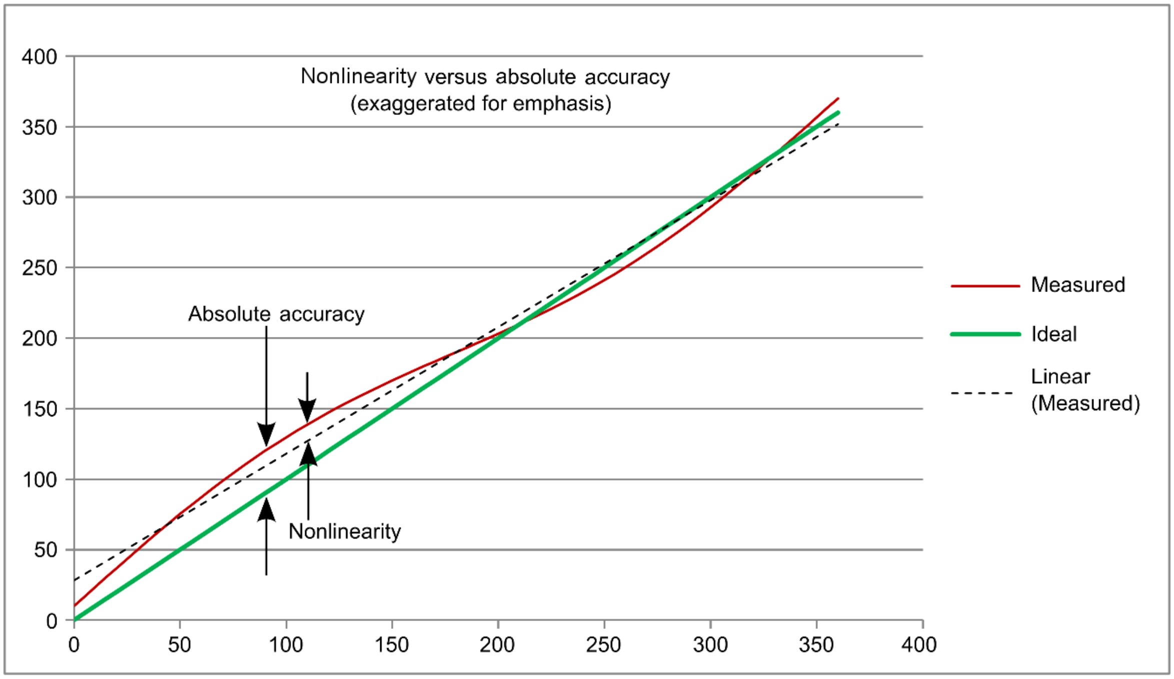
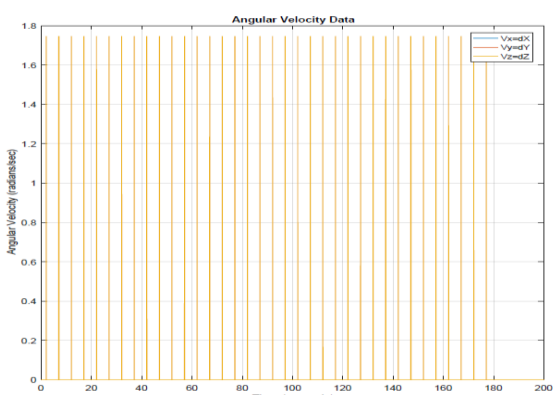


Figure 18. Nonlinearity vs. absolute accuracy

5.4.2 **Procedure**

Setup as defined in “Simulation Environment”.

1. Select soft iron model ranging from no soft iron distortion up to 3:1 in any of X, Y or Z directions.
2. Precondition the DUT by executing magnetic calibration trajectory leaving DUT in default R0 orientation.
3. Rotate DUT around the Z-axis from 0 to 360 degrees with an increment step of 10 degrees as shown:



Angular Velocity Impulses to Drive Heading Accuracy Test

4. Plot ideal versus simulated Yaw for each algorithm of interest

5.4.3 **Results**

Table 14 provides some basic guidance with regard to performance of the magnetic calibration library in a stand-alone configuration.

Table 14. Magnetic calibration performance metrics

Characteristic	Symbol	Conditions	Min	Typ	Max	Unit
Compass heading linearity	CH_{lin}	—	—	< 5	—	degrees
Compass heading accuracy	CH_{acc}	—	—	< 5	—	degrees
Maximum soft iron distortion	SID_{max}	Version 7.00 Version 7.10		1.5:1 3:1		unitless
Note: *The CH_{lin} and CH_{acc} results shown in this table are more conservative than simulated numbers, and are more likely to be representative of actual results.*						

5.4.4 **eCompass Considerations**

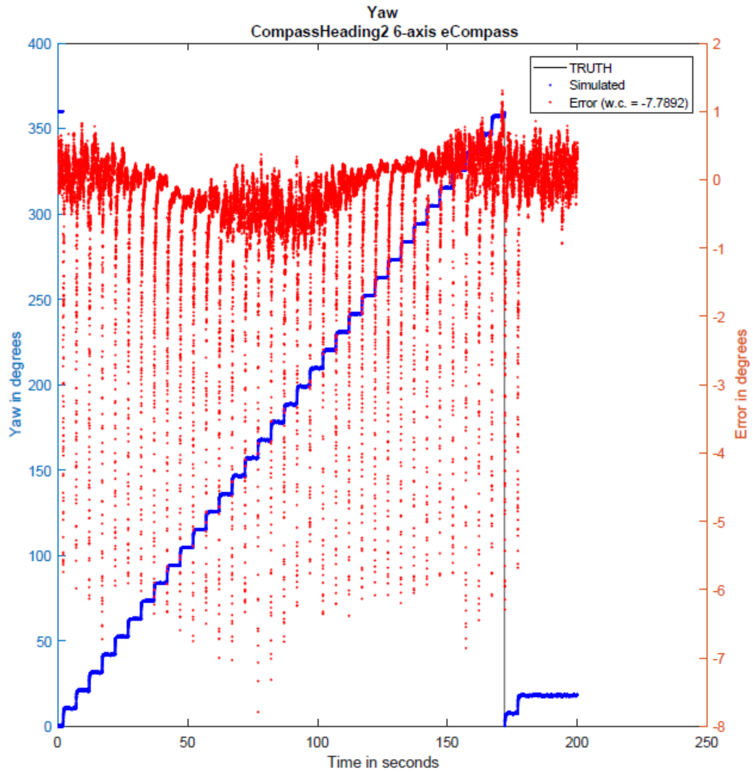


Figure 19: Simulated eCompass Heading

The plot above shows actual and simulated compass heading for a simulation of 3:1 soft iron distortion in X. Each step corresponds to a movement of 10 degrees. Error spikes at each transition are a natural result of the low pass filter incorporated into the eCompass algorithm. One can easily see that the steady state error is actually in the range of +/- 1 degree for this simulation.

5.4.5 **9-axis Compass Heading**

The figure below shows the results from several compass heading simulations utilizing the 9-axis Kalman filter. The major error appears to be a random walk of magnitude +/- 2.5 degrees or less.

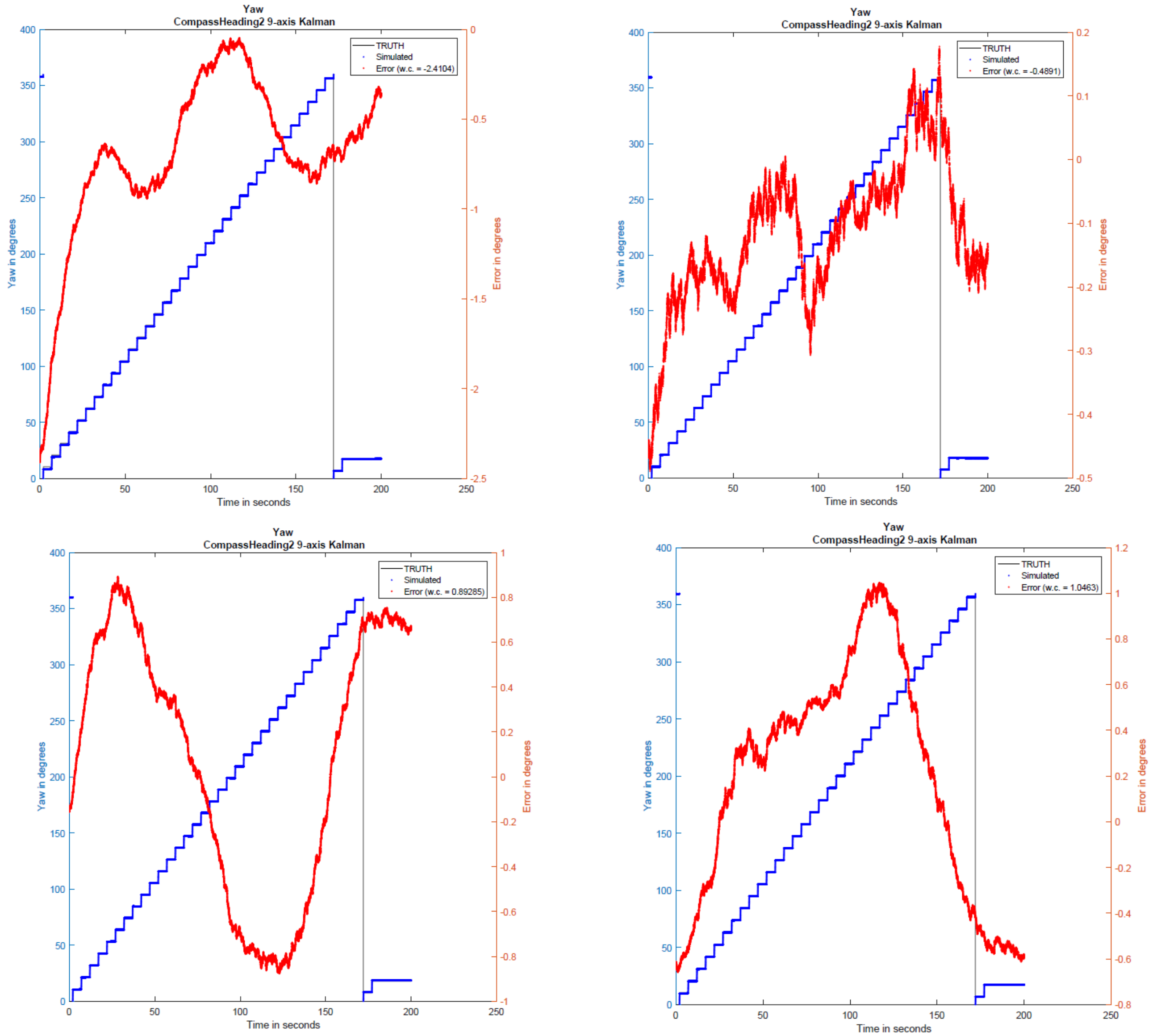


Figure 20: 9-axis Kalman Compass Simulations

5.5 Orientation Error Sensitivity to Magnetic Interference

5.5.1 **Static Device / Moving Magnet**

5.5.1.1 Intent

The test response of the stationary DUT to momentary changes in the local magnetic field is used to measure the orientation magnetic immunity. Because the device is stationary, the accelerometer and gyroscope readings remain relatively constant, changing only due to sensor noise.

5.5.1.2 Procedure

Setup as defined in “Simulation Environment”.

1. Starting point = device stationary, fusion outputs stable
2. 100 μT magnet moving at 0.25 m/s
3. Closest approach to magnetic sensor = 5 cm
4. The magnetic field is modeled using simplified expressions for a coil-generated field.

The figure below illustrates two events:

- Between time zero and one second, the DUT it rotated to 1 of 6 standard orientations.
- An external magnetic source is moved at 0.25 m/s in a line parallel to, and 5 cm above the DUT. The interfering field is 100 μT at the closest approach at time = 4 seconds.

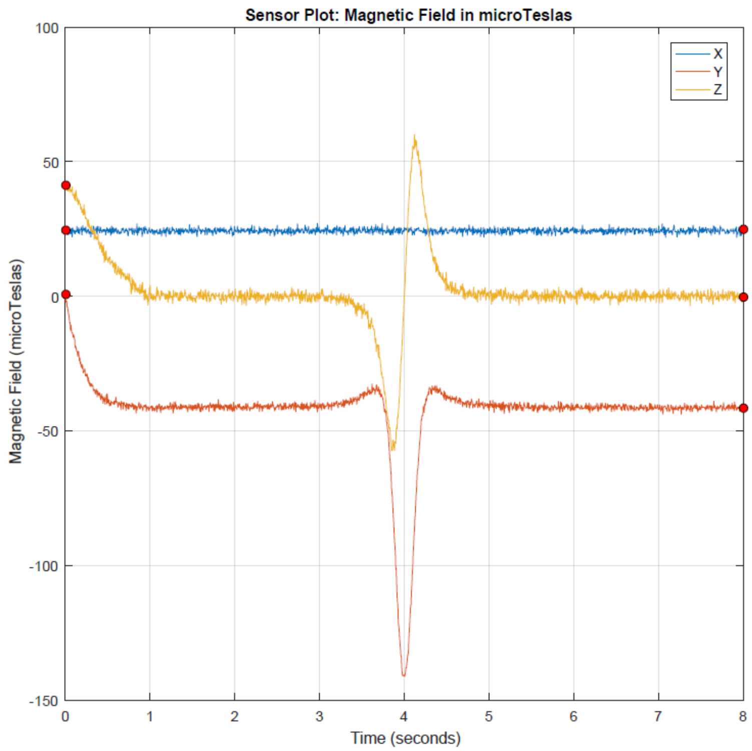


Figure 21: Magnetic Fields vs Time

5.5.1.3 Results

Simulated errors resulting from the magnetic interference are:

Table 15: Static Device / Moving Magnet Interference

Algorithm	Error
6-axis eCompass	63 degrees
6-axis accel/gyro Kalman Filter	Not Applicable
9-axis Kalman filter	~ 0.6 degrees

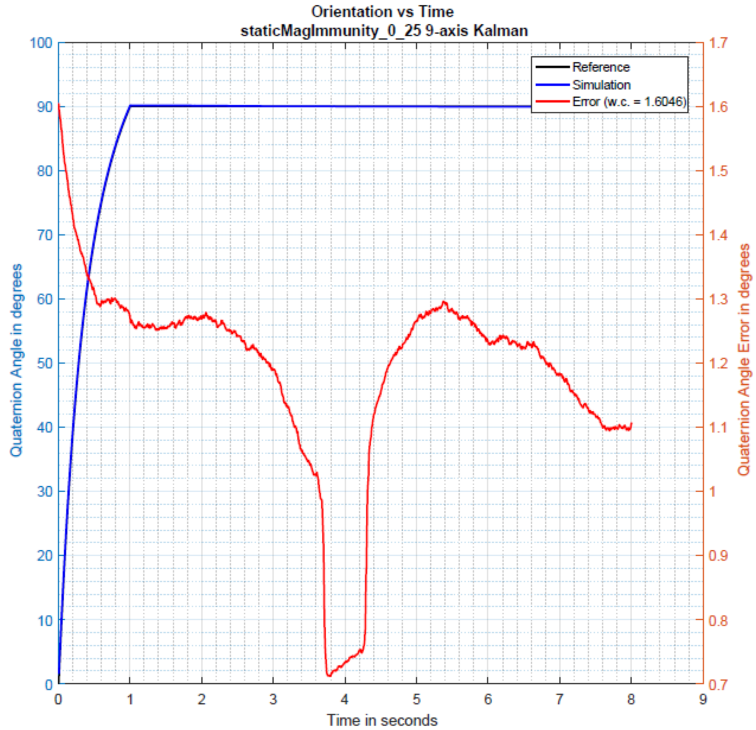


Fig 22: Example Output for Static Magnetic Immunity Simulation

5.5.2 **Moving Device / Static Magnetic Field**

5.5.2.1 Intent

This test measures the immunity of a linearity, with no rotation, moving DUT relative to a 100 µT magnet change in the magnetic field. The outputs of all acceleration and magnetic sensors change during this test. Gyro outputs should be constant throughout, with any changes attributed to noise only.

5.5.2.2 Procedure

Setup as defined in “Simulation Environment”.

1. Starting point = device stationary, fusion outputs stable.
2. Use a 100 µT magnet.
3. The DUT moves by magnet at 0.25 m/s with the closest approach to magnet = 5 cm.
4. The magnetic field is modeled using simplified expressions for a coil-generated field.

5.5.2.3 Results

Dynamic magnetic immunity tests, where the magnet is stationary and the DUT moves, yield results similar in magnitude to those shown for the static tests.

5.6 Orientation Error Sensitivity to Linear Acceleration

5.6.1 Intent

Depending upon the parameter, filter response to physical position and orientation changes may have a strong dependency on the actual movement being simulated. This series of tests is designed to explore those dependencies.

5.6.2 Procedure

Setup as defined in “Simulation Environment”.

A series of simulations were run with two slightly different acceleration profiles. In each case, orientation is constant and aligned to the global frame. One profile uses a single cycle of a sin wave, the other uses a single cycle of a square wave. Both had an amplitude of +/-2 g in X, 0g in both Y and Z. Simulations were run with periods of 0.5, 1.0 and 5.0 seconds.

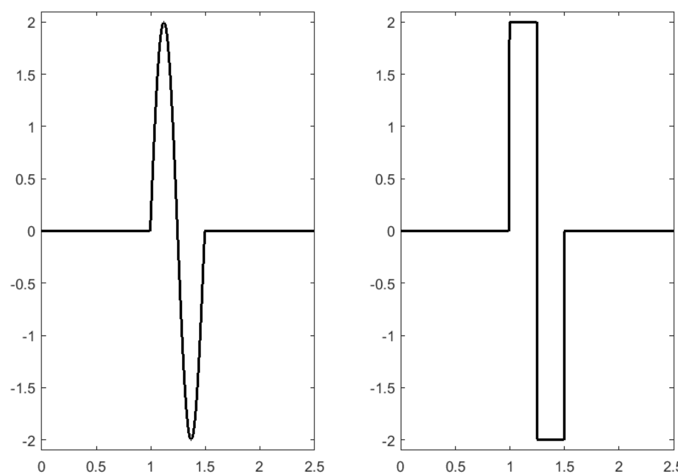


Figure 23: Alternate +/-2 g in X dimension stimulus sets

5.6.3 Results

The 9-axis filter is indeed relatively immune to orientation errors due to linear acceleration. Visible changes for these simulations are < 0.1 degrees (most of the error in Figure 24 appears to random offset drift).

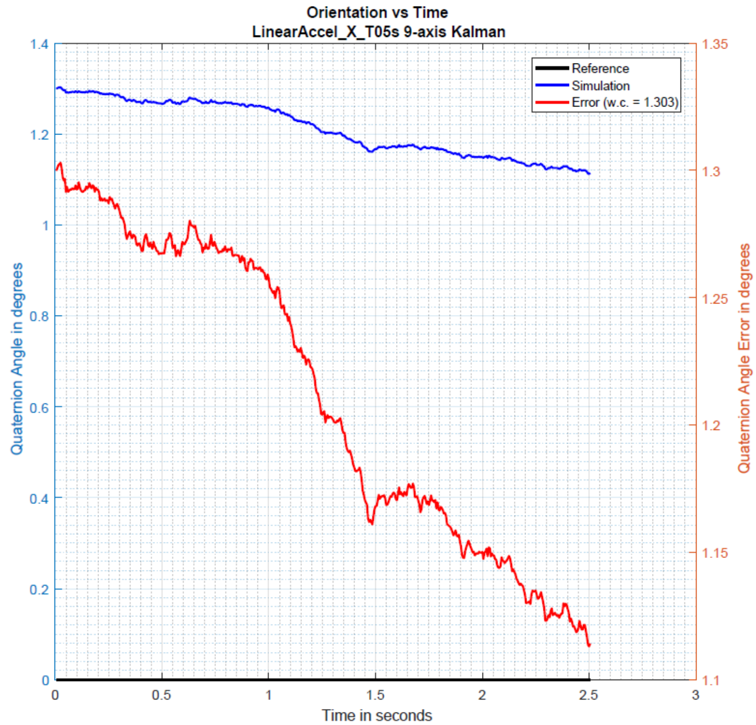


Figure 24: 9-axis Kalman Orientation Error for 0.5 second period sin wave

The 6-axis accel/gyro Kalman filter however does show errors for the left (sin wave) case, but not for the right. For the sin wave, we have the response in orientation shown in Figure 25, which you will recall should be unchanging.

You can see that the single cycle sin wave has resulted in a significant error in the orientation estimate.

Conversely, the square wave (Figure 26) has <1 degree effect on orientation result. The theory to explain the two different sets of results is that for the square wave case, usually the value of acceleration for a given timepoint is exactly the same as the previous timepoint. There are only 3 points in time at which there are discontinuities. Since the Kalman filter uses extrapolation to estimate new points from old, it should be very accurate for the square wave.

For the sin wave, there are always differences from one time point to the next during the sin wave, hence there will be estimation errors.

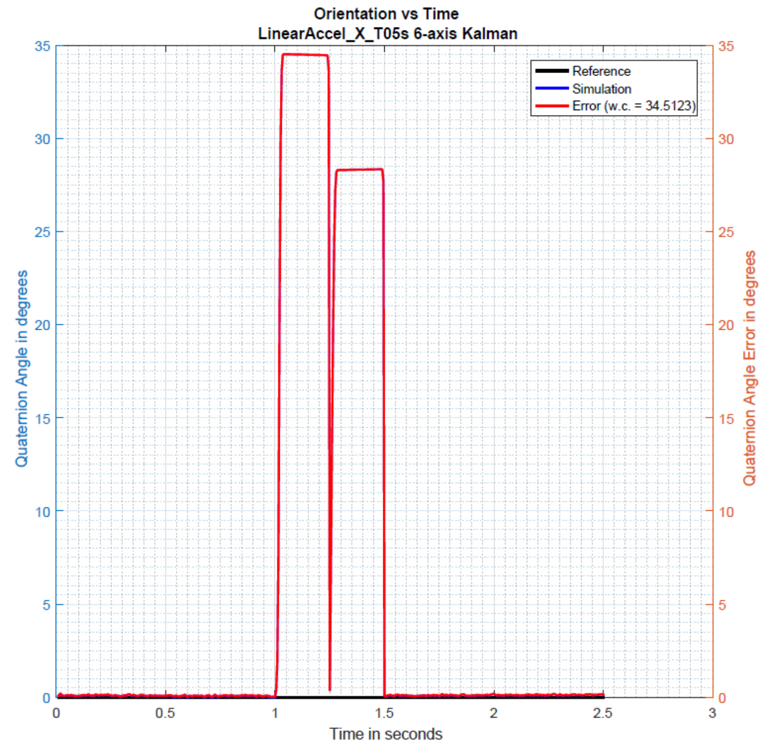


Figure 25: 6-axis Accel/Gyro Kalman Filter Orientation Response to Single Cycle X Acceleration Sin Wave

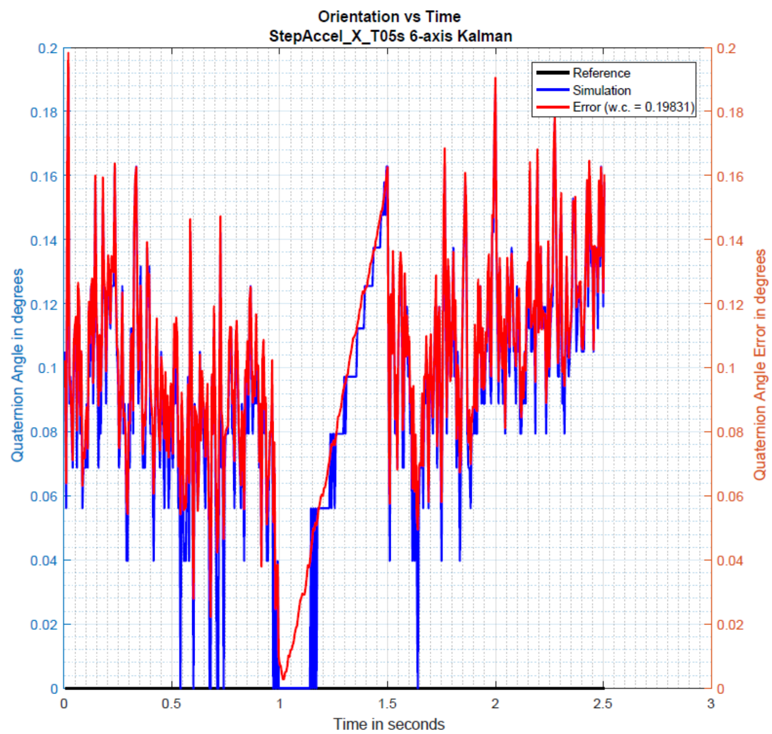


Figure 26: 6-axis Accel/Gyro Kalman Filter Orientation Response to Single Cycle X Acceleration Square Wave

Relative frequency can also have an effect. We simulated three different sin wave periods: 0.5, 1.0 and 5.0 seconds. The maximum orientation error seen for each is shown in the table below.

Table 16: Maximum 6-axis Kalman Orientation Errors as a function of +/-2 g sin wave

Sin Wave Period	Maximum Orientation Error
0.5 seconds	34.5 degrees
1.0 seconds	22.2 degrees
5.0 seconds	12.8 degrees

5.7 Errors in Linear Acceleration Estimates

5.7.1 Intent

Linear acceleration is computed by translating measured gravity-acceleration (assuming NED frame of reference) from the sensor frame to the global frame, and then subtracting the gravity component. Thus, any error in the orientation estimate directly affect the linear acceleration estimate.

5.7.2 Procedure

Setup as defined in “Simulation Environment”.

The same simulation used for orientation sensitivity to linear acceleration is used here.

5.7.3 Results

The two figures below represent the horizontal and vertical components of acceleration for the same simulation shown in Figure 23.

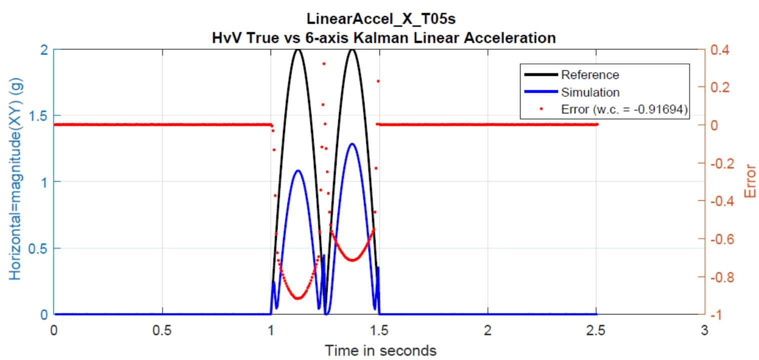


Figure 27 : 6-axis Kalman Horizontal Acceleration Response to Single Cycle X Acceleration Sin Wave

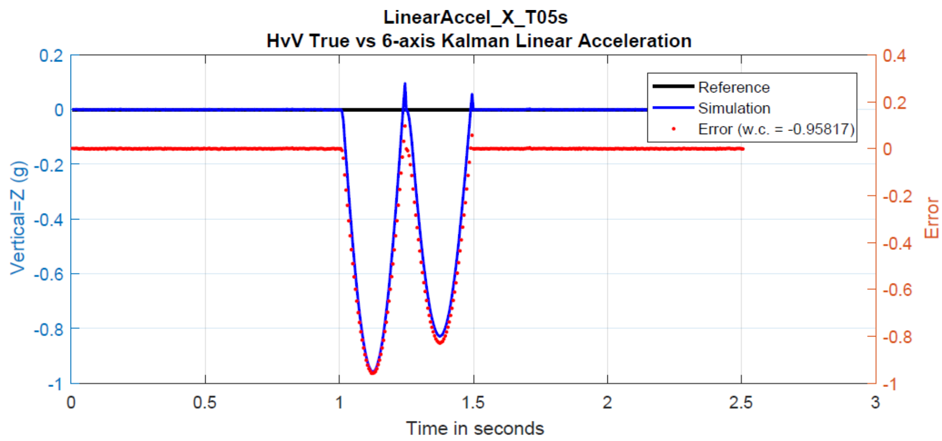


Figure 28. 6-axis Kalman Vertical Acceleration Response to Single Cycle X Acceleration Sin Wave

Expanding Table 16 to include acceleration estimate errors, we see the direct correlation between orientation errors and errors in computed acceleration.

Table 17: Maximum 6-axis Kalman Orientation Errors as a function of +/-2 g sin wave

Sin Wave Period	Maximum Orientation Error	Acceleration Error in the Horizontal Direction	Acceleration Error in the Vertical Direction
0.5 seconds	34.5 degrees	0.92 g	0.96 g
1.0 seconds	22.2 degrees	0.53 g	0.68 g
5.0 seconds	12.8 degrees	0.27 g	0.41 g

Finally, the next figure looks at acceleration errors out of the 9-axis Kalman filter. Errors are less than 10mg for each of X, Y and Z axes.

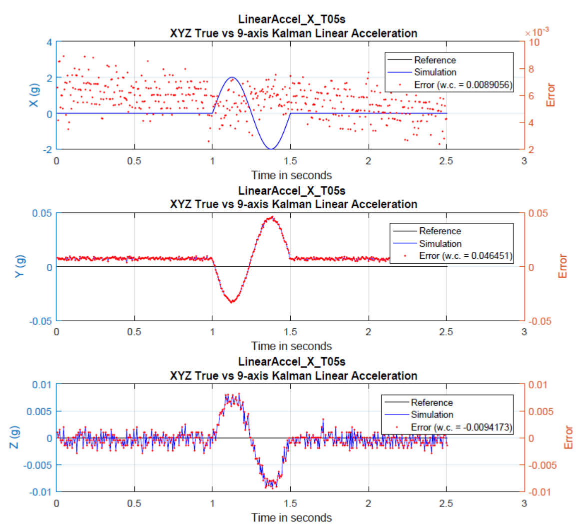


Figure 29: 9-axis Kalman acceleration errors for the 0.5s 2g sin wave example

5.8 Orientation Response Delay

5.8.1 Intent

Determine the amount of time required for any of the filters to respond to a change in device orientation.

5.8.2 Procedure

Setup as defined in “Simulation Environment”.

Prior versions of this document measured this parameter using a 90 degree orientation change and from the 50% point on the input to 50% point on the output waveform. This version instead seeks out the phase delay which results in optimal cross-correlation between the waveforms.

The DUT is subjected to 10 ms pulses of +/- 450 dps in angular velocity. This results in orientation changes of +/- 45 degrees.

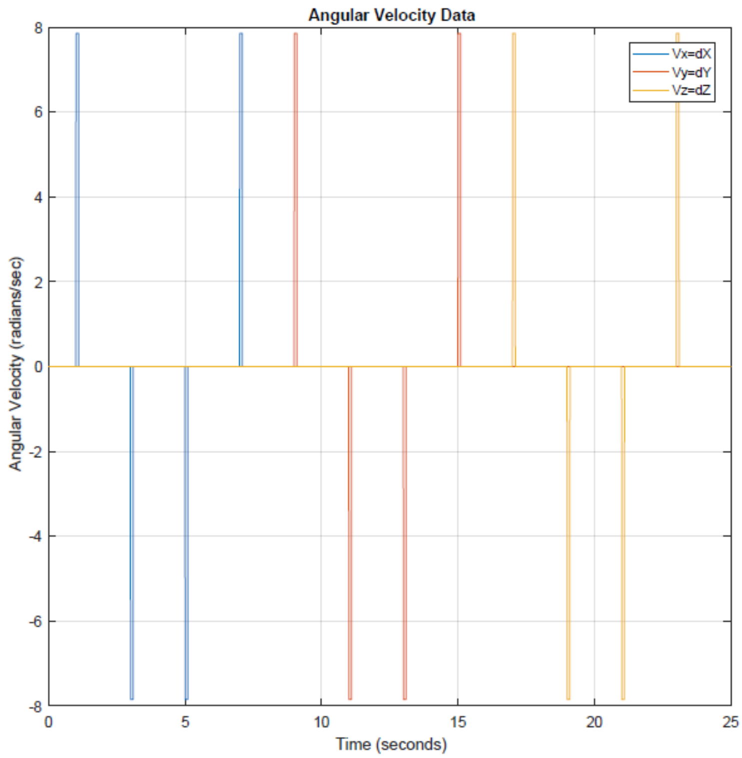


Figure 30: Angular Velocity Input Pattern

Orientation response delays were computed using the following Matlab snippet of code:

```
[acor, lag] = xcorr(IdealAngle, ComputedAngle);
 [~,I] = max(abs(acor));
timeDiff = lag(I)/sps;
```

5.8.3 Results

Table 18: Computed Input/Output Phase Delays

Filter	Phase Delay
eCompass	30 ms
6-axis Accel/Gyro Kalman	0 ms
9-axis Kalman	0 ms

At sps = 200 samples/second, which is also the fusion rate for this testcase, the eCompass delay equates to 6 cycles of the filter. You can see the characteristic low pass response curves in the figure below.

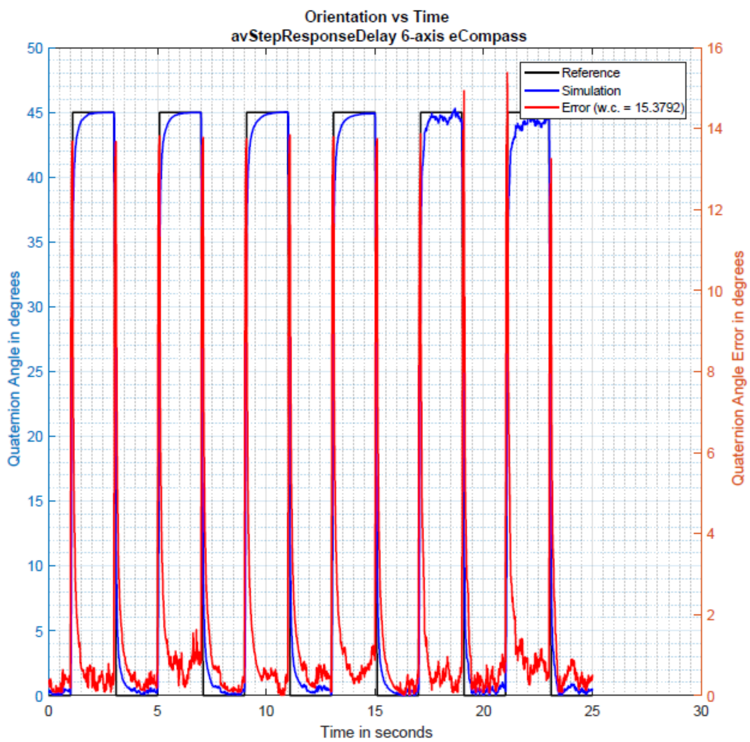


Figure 31: eCompass Response

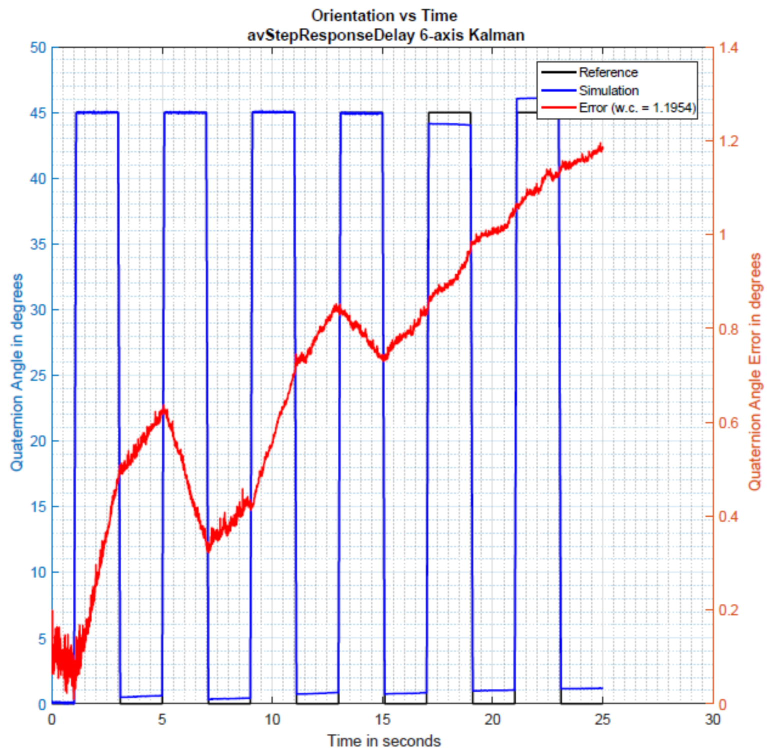


Figure 32: 6-axis accel/gyro Kalman Response

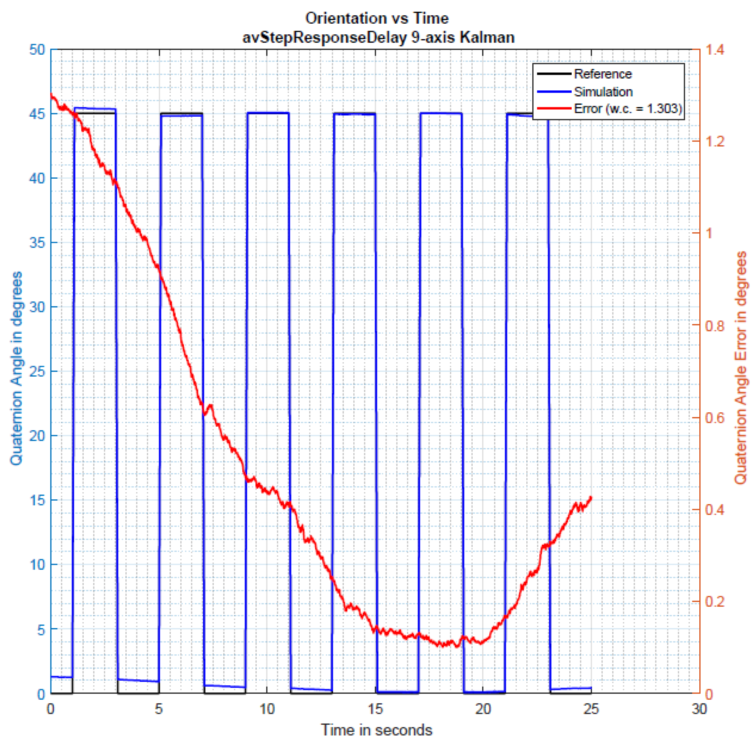


Figure 33: 9-axis Kalman Response

The zero cycle delays of the Kalman filter relate to the fact that Kalman filters make both a priori and a posteriori estimates of the output signal. So with a regular waveform such as those used here, the computed output will naturally lie on top of the input.

For arbitrary waveforms, NXP recommends you assume 2 fusion cycles of delay. One cycle represents sensor pipeline delay, and the second is the Kalman filter delay.

5.9 Maximum Angular Rate

5.9.1 Intent

To determine whether any of the algorithms have a hard upper limit on angular rate inputs.

5.9.2 Procedure

Setup as defined in “Simulation Environment”.

Ramp angular rate from 0 to 2000 dps. Compare input and output waveforms.

5.9.3 Results

There are no intrinsic angular rate limitations in either 6- or 9-axis Kalman filter. The figure below shows that errors are on the range of +/- 1 dps. Striations shown in the error data are a result of ADC quantization.

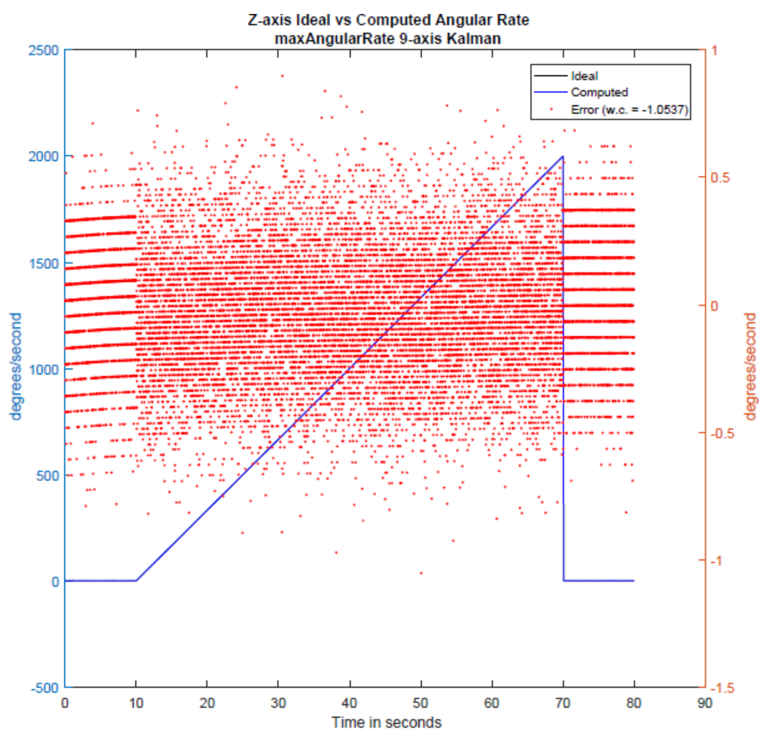


Figure 34: Actual vs Computed Angular Rate in Z

Angular rate will have an effect on eCompass accuracy. Figure 35 on the next page shows errors in yaw increasing to approximately 25 degrees at an angular rate in Z of 2000dps, then dropping back near zero when the rotation ceases.

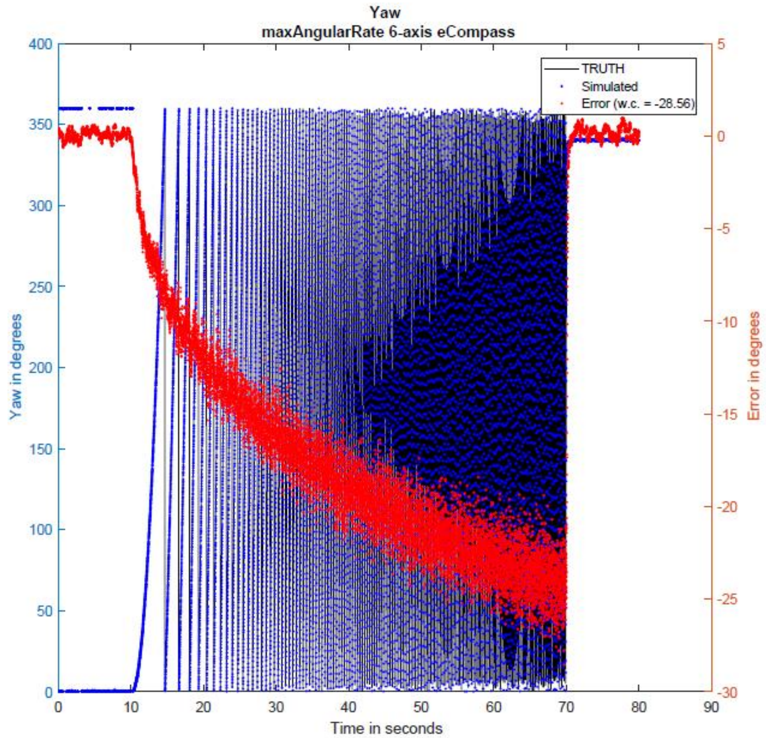


Figure 35: eCompass Sensitivity to Angular Rate in Z

5.10 Orientation dynamic drift

5.10.1 **Intent**

Measure of the ability of fusion code to return to a known orientation after movement.

5.10.2 **Procedure**

Setup as defined in “Simulation Environment”.

1. Precondition the DUT by executing magnetic calibration trajectory, leaving DUT in a given orientation.
2. Keep DUT in a given orientation motionless for 2 seconds.
3. Rotate DUT at 0.5 revolution/sec for 10 seconds around vertical axis.
4. Keep DUT in its last orientation motionless for 2 seconds.
5. Determine the initial orientation by averaging orientation samples from the initial motionless period and the final orientation by averaging the samples from the final motionless period. The difference between the two orientations is the orientation dynamic drift.
6. Repeat steps 1–5 for all six major orientations of the development board.

5.10.3 **Results**

Table 19: Orientation Dynamic Drift Results

Algorithm	Orientation Dynamic Drift
eCompass	1.0 degrees

Algorithm	Orientation Dynamic Drift
6-axis Kalman	3.7 degrees
9-axis Kalman	1.2 degrees

5.11 Orientation Static Drift & Orientation Noise

5.11.1 Intent

Measure of the ability of fusion code to remain at a stable orientation while motionless.

Note: This test procedure has been modified from that used in prior versions of this document. The period has been shortened to the same as that used for the Orientation Dynamic Drift measurements.

5.11.2 Procedure

Setup as defined in “Simulation Environment”.

1. Precondition the DUT by executing magnetic calibration trajectory, leaving DUT in a given orientation.
2. While the DUT remains in given orientation, collect orientation samples for an additional 100 seconds.
3. Plot and process the last 100 s of the test. Orientation static drift is the maximum angle change in the measured rotation vectors.
4. Compute orientation noise over the same 100 second period.
5. Repeat steps 1–4 for all six major orientations of the development board.

5.11.3 Results

Table 20: Orientation Static Drift Results

Algorithm	Orientation Static Drift (over 100 seconds)	Orientation Noise
eCompass	1.8 degrees	0.16 degrees
6-axis Kalman	30.4 degrees	1.62 degrees
9-axis Kalman	.4 degrees	0.06 degrees

It is important to note that the simulator used to compute these numbers did not include the “Fusion Standby” mode discussed in Section 4.9 of the Sensor Fusion User Guide. Use of that feature allows the fusion engine to completely shut down after a few seconds of inactivity, as measured by the system accelerometer. This will dramatically reduce the 6-axis Kalman static drift numbers.

5.12 Summary Mechanical Results

5.12.1 9-Axis

Table 15 provides guidance for the 9-axis (accelerometer/magnetometer/gyro) indirect Kalman filter implementation.

Note: *This datasheet is intended to characterize NXP sensor fusion algorithms. Accordingly, sensor models include 1st order effects only. Physical sensor non-linearities and misalignments may negatively effect parameters shown below. All parametrics provided in the following table are based on Version 7.00 simulations utilizing models described in "Simulation Environment".*

Table 21. 9-axis Sensor Fusion performance metrics

Characteristic	Symbol	Section Reference	Min	Typ	Max	Unit
Orientation static drift	O_{SD}	5.11	—	0.4	—	degrees
Orientation static noise	O_{SN}	5.11	—	0.1	—	degrees RMS
Orientation dynamic drift	O_{DD}	5.10	—	1.2	—	degrees
Max angular Rate	AR_{MAX}	5.9	—	2000[1]	—	dps
Orientation response delay	O_{RD}	5.8	—	< 2	—	fusion time interval
Gyro offset step response	T_{GOSR}		—	TBD	—	seconds
Error in computed linear acceleration in each of X, Y & Z dimensions	LAE	5.7	—	10	—	*mg*
Compass heading linearity[2]	CH_l	5.4	—	5	—	degrees
Compass heading accuracy	CH_{acc}	5.4	—	5	—	degrees
Orientation magnetic immunity - static device	O_{mis}	5.5.1	—	0.6	—	degrees
Orientation magnetic immunity - moving device	O_{mim}	5.5.2	—	0.6	—	degrees

1. The Sensor Fusion algorithm has no intrinsic limitation; this was the maximum value supported by the gryo in NXP testing.
2.. Linear sensors, which yields very good compass heading values were assumed. However experience shows that ±5 degrees are more realistic values.

5.12.2 6-Axis eCompass

Note: *This datasheet is intended to characterize NXP sensor fusion algorithms. Accordingly, sensor models include 1st order effects only. Physical sensor non-linearities and misalignments may negatively effect parameters shown below. All parametrics provided in the following table are based on Version 7.00 simulations utilizing models described in "Simulation Environment".*

Table 22. 6-axis Sensor Fusion accel + mag performance metrics

Characteristic	Symbol	Section Reference	Min	Typ	Max	Unit
Orientation static drift	O_{SD}	5.11	—	1.8	—	degrees
Orientation static noise	O_{SN}	5.11	—	0.16	—	degrees RMS
Orientation dynamic drift	O_{DD}	5.10	—	1.0	—	degrees
Orientation response delay	O_{RD}	5.8	—	30	—	ms[1]
Compass heading linearity[2]	CH_l	5.4	—	5	—	degrees
Compass heading accuracy[2]	CH_{acc}	5.4	—	5	—	degrees

1. At fusion frequency of 200Hz. This equates to 6 fusion time intervals.
2. Linear sensors, which yields very good compass heading values were assumed. However, experience shows that ±5 degrees are more realistic values.

5.12.3 **6-axis Kalman**

Note: *This datasheet is intended to characterize NXP sensor fusion algorithms. Accordingly, sensor models include 1st order effects only. Physical sensor non-linearities and misalignments may negatively effect parameters shown below. All parametrics provided in the following table are based on Version 7.00 simulations utilizing models described in "Simulation Environment".*

Table 23. 6-Axis Sensor Fusion gyro + accel performance metrics

Characteristic	Symbol	Section Reference	Min	Typ	Max	Unit
Orientation static drift	O_{SD}	5.11	—	30.4	—	degrees
Orientation static noise	O_{SN}	5.11	—	0.16	—	degrees RMS
Orientation dynamic drift	O_{DD}	5.10	—	3.7	—	degrees
Max angular rate	AR_{MAX}	5.9	—	2000	—	dps
Orientation response delay	O_{RD}	5.8	—	< 2	—	Fusion time intervals
Gyro offset step response	TBD		—	3.76	—	seconds

5.12.4 **3-Axis Accelerometer Only Results**

Table 24. 3-Axis accelerometer performance metrics

Characteristics	Symbol	Min	Typ	Max	Units
Tilt Error RMS[1]	TAE	—	0.082	—	degrees
Orientation response delay	O_{RD}	—	< 6	—	Fusion time intervals

1. RMS of accelerometer tilt angle error is calculated using the simulated sensor noise RMS values along each of the three axes and the following formula (given for tilt error from the Z-axis)

$$\mathrm{TAE} = \arctan\left(\frac{\sqrt{\mathrm{NRMS^2}_x + \mathrm{NRMS^2}_y}}{1g - \mathrm{NRMS}_z}\right)$$

where NRMSx, NRMSy, NRMSz - RMS values of accelerometer noise for X, Y, Z axes in *g* units.

6 Revision history

Rev. No.	Date	Description
0	12 Nov 2013	• ROUGH DRAFT ONLY - PRE-REVIEW
0.1	22 Nov 2013	• Preliminary draft includes updates from 1st review.
0.2	Feb 2014	• Initial public release.
0.3	Apr 2014	• Updated for licensing, software updates and board support changes.
0.4	May 2014	• Updated for software updates, additional (FRDM-K64F) board support changes and electrical specs and computation metrics.
0.5	Sept 2014	• Updated Fusion Performance Metrics section by adding four new figures and tables. • adjusted selected parametric values • altered several Test Description procedures.
0.6	Sept 2014	• Separated out Computational Metrics section and various minor markups • Changed Feature - License, option text. • Feature Comparison Based on License Option, Added KDS to Product Deliverables row • Moved sections 4.1, 4.1.1 & 4.1.2 and merged in 4.11 • Added xrefs from Electrical Specs tables to appropriate Test Description sections • Adjusted Performance Metric tables, symbols and units in some cases
0.7	Sept 2015	• Updated for build 5.00 of the sensor fusion library. This version has completely redesigned 6 and 9-axis Kalman filters. • Updated all computation metrics. • Removed outdated fusion time measurements. • Added K22F for KDS. Added additional fusion options. • Changed document name from XSFLK_DS to NSFK_DS. • Noted that MULTI-B boards have been replaced with MULT2-B boards.
0.8	August 2016	• Document updated to reflect change from Freescale to NXP. • Compute parameters and IDD measurements updated to match sensor fusion Version 7.00.
0.9	November 2016	• Major restructuring of the document. Kalman performance metrics have been simulated using a Version 7.xx-based simulation tool. Procedures updated accordingly.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number: NSFK_DS
Revision Rev. 0.9
11/2016