The Limits of In-Run Calibration of MEMS Inertial Sensors and Sensor Arrays

HENRY MARTIN and PAUL GROVES
University College London, London, UK

MARK NEWMAN
BAE Systems Applied Intelligence Laboratories, Chelmsford, UK

Received March 2015; Revised February 2016

ABSTRACT: *MEMS accelerometers and gyroscope triads now cost less than $10, potentially opening up many new applications. However, these sensors require calibration prior to navigation use.*

This paper determines the maximum tolerable sensor errors for in-run calibration techniques using a basic Kalman filter by developing criteria for filter failure and performing Monte Carlo simulations for a range of different sensor specifications, and both car and UAV motion-profiles. Gyroscope bias is found to be the most significant with the maximum tolerable value of its SD varying between 0.75 and 2.6 deg/s depending on the value of the specification of the other sensor sources. The paper shows that pre-calibration and smart array techniques could potentially enable in-run calibration to be applied to lower-quality sensors. However, the estimation of scale-factor cross-coupling and gyroscope g-dependent errors could potentially be critical.

Armed with this knowledge, designers can avoid both unnecessary design complexity and computational load of over-engineering and the poor navigation performance of inadequate filters. Copyright © 2016 The Authors Journal of the Institute of Navigation published by Wiley Periodicals, Inc. on behalf of Institute of Navigation.

INTRODUCTION

Inertial sensors can significantly increase the robustness of an integrated navigation system by bridging gaps in the coverage of other positioning technologies, such as GNSS, Wi-Fi, and various environmental feature-matching techniques [1]. A full set of chip-scale micro-electro-mechanical sensors (MEMS) accelerometers and gyroscopes can now be bought for less than $10 (e.g., Invensense MPU-9150 '9-axis' IMU (+magnetometer) available from www.digikey.co.uk from £4.40, MPU-6050 '6-axis' IMU from £3.87), potentially bringing inertial and integrated navigation to a wide range of new applications. However, these sensors are uncalibrated, exhibiting large temperature-dependent biases and scale factor errors. Before they can be used for navigation, some form of calibration is required [2].

An inertial measurement unit (IMU) with relatively small biases and other systematic errors can be calibrated during normal use using a second navigation system, such as GNSS and a basic Kalman filter (KF) integration algorithm [1, 3–5]. This enables the IMU's calibration to be frequently updated without any effort by the user, whenever another navigation technology is available. However, when attempting to calibrate larger errors the KF-integration starts to break because linearity and small angle approximations made within its system model are not valid.

One solution is to replace the Kalman filter with a nonlinear estimation algorithm, such as an unscented Kalman filter [6] or a particle filter [7]. Other techniques suggested to improve filter stability include adaptive filtering [8–11] and automatic tuning [12]. However, all of these increase the processing load. Other possible solutions include a pre-calibration procedure and a smart array. Pre-calibration requires the user to perform a known series of maneuvers, which enable a deterministic algorithm to perform a coarse calibration of the sensor errors prior to the initialization of the integration Kalman filter. A smart array is an IMU comprising an array of inertial sensors that have been arranged to exploit the design characteristics of the sensors such that the errors exhibited are smaller than those obtained by simple averaging [13].

NAVIGATION: *Journal of The Institute of Navigation*
Vol. 63, No. 2, Summer 2016
Printed in the U.S.A.

In order to establish when these alternatives are necessary, we must first determine the limits of the normal Kalman filter. Armed with this knowledge, the community can avoid both the unnecessary design complexity and computational power consumption caused by over-engineering the filter and the poor navigation performance that arises from an inadequate filter. After establishing realistic limits, one can determine whether real sensors are suitable for in-run calibration with simple characterization tests, rather than having to perform time-consuming empirical testing.

In this paper, we will establish where these limits are. First, we present some background on inertial measurement units, user-conducted calibration procedures and Kalman filter-based INS/GNSS integration (*Background* section). Then we establish well-defined filter failure criteria (*Kalman Filter Failure* section). These are crucial for determining when the current filter is inadequate and a more complex filter is needed.

Having established suitable failure and success criteria, Monte Carlo simulations are performed with a range of different sensor specifications in order to determine the maximum tolerable sensor errors. The approach to computing these simulations is presented in the *Simulation Approach* section, and the results of these simulations with a car and quadcopter motion profiles are shown in the *Results* section. The implications of the research are discussed in the *Practical applications of the results* section, and the conclusions are drawn in the *Conclusions* section.

A preliminary version of this study was presented at the 2014 ION GNSS+ conference [14]. Here, a new approach to tuning the filter is presented (*Filter Tuning* section), so all results are new. A quadcopter motion scenario has also been added to the car motion scenario from [14].

BACKGROUND

Inertial Measurement Units and their Errors

An IMU is a set of (at least) three accelerometers and three gyroscopes, arranged so that specific force and angular rate can be measured in all three dimensions. Typically, the sensitive axes are mutually orthogonal. An INS uses the output of an IMU to calculate a dead-reckoning navigation solution [1, 15]. The attitude (3D orientation) is computed by integrating the angular rate measurements. Then, an estimate of Earth's gravity is added to the specific force measurements to provide acceleration, which is integrated twice to calculate the change in position. Thus, large position errors can arise from small errors in the specific force measurements. Additionally, as the attitude solution is used to determine the correct direction of Earth's gravity, small errors in the attitude solution can also produce large position errors. Thus, accurate measurement of both specific force and angular rate is very important in maintaining a good navigation solution.

Inertial sensor errors can be split into two categories: stochastic and systematic. Stochastic errors are random in nature; the most significant is usually the random noise added to the signal. If the noise is predominantly white, then improvement can only be made by sacrificing bandwidth, as white noise cannot be removed by calibration. If the stochastic errors are too high, the only solution is to use more or better quality sensors.

Systematic errors are fixed and/or a function of the sensors input and the environmental conditions. If these errors can be measured, their effect can be compensated for in the IMU's output. Systematic errors include bias, scale factor error, cross-axis sensitivity, and gyroscope g-dependent error. Bias is a fixed error. Scale factor error is when the sensitivity of the sensor is incorrect, such as measuring 98% or 101% of the true quantity. Cross-coupling is sensitivity to specific force or angular rate in directions perpendicular to the nominal sensitive axis. One case of this is mounting misalignment. Gyroscope g-dependent error is when a gyroscope is sensitive to specific force. There are other higher-order errors such as sensor nonlinearity, about which there is more information in [1, 15, 16].

The magnitude of all of these errors will vary considerably between individual sensors of a particular model, as well as slightly from day-to-day and slowly during use. In addition, many of these errors also vary as a function of environmental conditions, particularly temperature. As such, the accuracy of an IMU calibration may reduce over time.

User Conducted Calibration

There are two main ways in which end-user calibration can be achieved. These are instruction based, where the user is given a specific set of maneuvers to carry out, and "in-run", where the inertial sensors are calibrated during the normal use of the navigation system.

The most important feature of instruction-based calibration is that it requires the user to stop using the system and to perform a series of maneuvers when, and precisely as, instructed, which may not be practical for all applications. The more complex this set of maneuvers is, the more of the systematic error sources can be determined, potentially allowing higher accuracy. However, as the maneuvers become more complex, the chance that the user performs them incorrectly increases. The simplest instruction-based calibration asks the user to 'place the IMU still on a table,' which allows the gyroscope bias to be measured. More complex sets of instructions could be used if the IMU is in a precisely cuboid box where static measurements on each of the six sides can be used to determine gyroscope and accelerometer bias, accelerometer scale factor, and gyroscope g-dependent error ([13] for details). If the user can perform more complex maneuvers (e.g., rotating slowly about particular axes in a specific order) or calibration equipment is available (e.g., rate-tables or temperature controlled chambers) even more, systematic errors could be determined [15].

In contrast, calibrating an inertial sensor's systematic errors 'in-run' relies on a second navigation system (e.g., GNSS) to provide measurements, which can be compared with those derived from the IMU, provided that a sufficient variety of maneuvers is conducted to make all the errors observable [1]. This means that while the second system is providing reliable measurements, the calibration can be performed continuously, so a much more recent calibration is available when the inertial-only navigation needs to be used. Moreover, no specific effort from the user is needed. However, there is one significant limitation. A reasonably good initial estimate of the systematic errors in addition to its uncertainty is required for the in-run estimation process to remain stable and converge towards the correct value.

INS/GNSS Integration Based on a Kalman-Filter-Derivative

Many different navigation technologies could be used to aid/calibrate an INS. Maximum robustness is acheived by combining many different sensors [17], but complex multisensor navigation brings many challenges [18]. In this paper, we assume that the INS is aided/calibrated using global satellite navigation systems (GNSS). This is the method most commonly used in practice due to the low cost of GNSS user equipment, free to use infrastructure, and fairly high accuracy and availability. INS/GNSS integration is a well-established technique [1, 4].

The Kalman filter (KF) [3] is an estimation algorithm that is linear; if the system is not, it must be linearized. A KF works in state space, that is, each quantity estimated is a state. For example, three states are needed for each of position, velocity, and attitude (for 3D navigation). When measurements are added to the filter, these measurements are compared with the measurements predicted by the current state estimates, projected to the current epoch, and used to correct the state estimates weighted according to the relative uncertainty of the measurements and the predicted state estimates. The filter is recursive, and in that, it does not store the old measurements; rather the state estimates aggregate this information in combination with the state error covariance matrix (the '**P**'-matrix), which represents the estimation accuracy of the filter's state estimates. More information about Kalman filtering can be found in standard texts such as [19, 20].

In order for the filter to behave correctly, the best possible initial estimates of the states should be used, and the state error covariance matrix should be correctly initialized with the 'initial uncertainty' of each state. The filter also needs to know how much the states are expected to change through time, and how accurate the measurements are. Choosing the correct level for all these settings is known as tuning, and often requires an element of trial-and-error.

The Kalman filter is based on the following assumptions [19]:

- the state estimation errors have a Gaussian (normal) distribution;
- the noise terms are 'white'-noise (not correlated with time); and,
- most importantly, the system propagation and measurements are linear combinations of states.

However, real systems, such as INS and GNSS, do not obey these rules. Linearizing approximations typically may include the 'small angle approximation' and the assumption that the products of state estimate errors are negligible. The white noise assumption inherent in Kalman filtering can be partially circumvented by telling the filter that the noise variance is greater than it really is (overbounding) in order to model noise that is time correlated over a few successive epochs [1].

If the these assumptions are not met or circumvented and/or the tuning is wrong, the Kalman filter will not behave as expected, e.g., estimates will not converge; there may even be numerical failure. In short, the Kalman filter will 'break' if the errors are 'too large.'

KALMAN FILTER FAILURE

To examine the limits of Kalman filter performance, we must establish criteria for when a Kalman filter is performing unacceptably, inadequately, or unstably, henceforth "failing".

In order to decide when a particular Kalman filter fails, we first need to examine how it should behave; we can then detect when failure has occurred.

A 'well-behaved' KF should start with state uncertainties, which slightly exceed the real standard deviation (SD) of the error distribution, to aid filter stability when the true stochastic behavior of the states diverges from the KF's assumptions. Then, as more aiding (e.g., GNSS) measurements are added, both the filter's state uncertainty and the real SD of the error in the state estimates should reduce together, particularly when maneuvers (such as turns) take place. This is illustrated in Figure 1. Note that the accuracy of the estimates eventually plateaus at a level that depends on the accuracy of the aiding measurements, the IMU noise, and the time variation of the systematic errors.

When the sensor errors are very large, the linearizing assumptions described in the *INS/GNSS integration based on a Kalman-filter-derivative* section break very quickly. For example, GNSS integration can normally only correct attitude errors indirectly based on their effect on the position/velocity solution. The KF uses the off-diagonal elements of the P-matrix to infer the attitude and instrument errors from the position and velocity errors. When the gyro biases are large, the attitude error can grow more quickly than it can be corrected, breaching the small-angle approximation. When the 'small angle' approximation ceases to be valid, the P-matrix no longer models the correlations between the states correctly, so the corrections to the state estimates from the measurements can be applied incorrectly.

When the errors are 'large,' further KF assumptions, such as product of two state-errors being negligible, can also break down. This kind of behavior can be seen in Figure 2. This produces similarly erratic estimates for the IMU error states, as shown in Figure 3. Clearly, if we run a simulation and the results are as erratic as in that example, detecting that this is a 'failure' is straightforward.

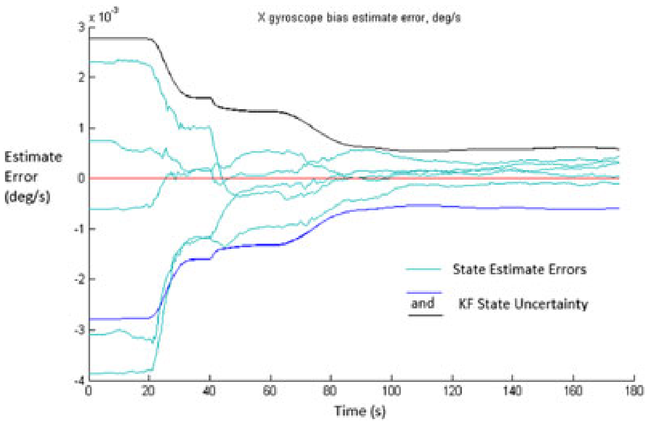


Fig. 1–This shows five runs of a Monte Carlo simulation to illustrate how a typical state estimate behaves when a KF integration is working as intended. In this example, the estimation errors of this state (X gyroscope bias) converge towards zero and their distribution is well described by the KF's state uncertainty.

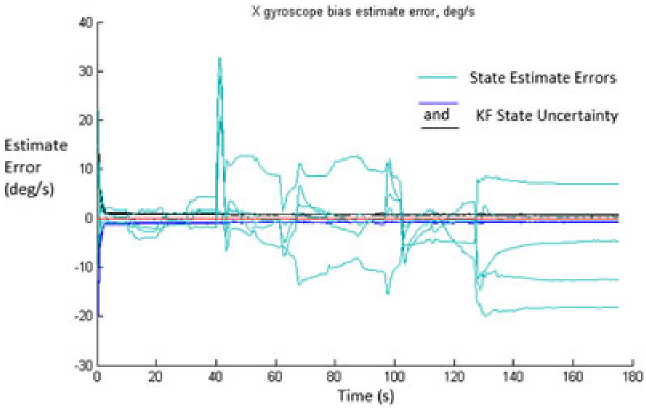


Fig. 3–This shows five runs of a Monte Carlo simulation to illustrate how a typical state estimate behaves when a KF integration is not working as intended. In this example the estimation errors of this state (X gyroscope bias) do not converge to zero and vary erratically, additionally their distribution is much greater than the KF's state uncertainty.

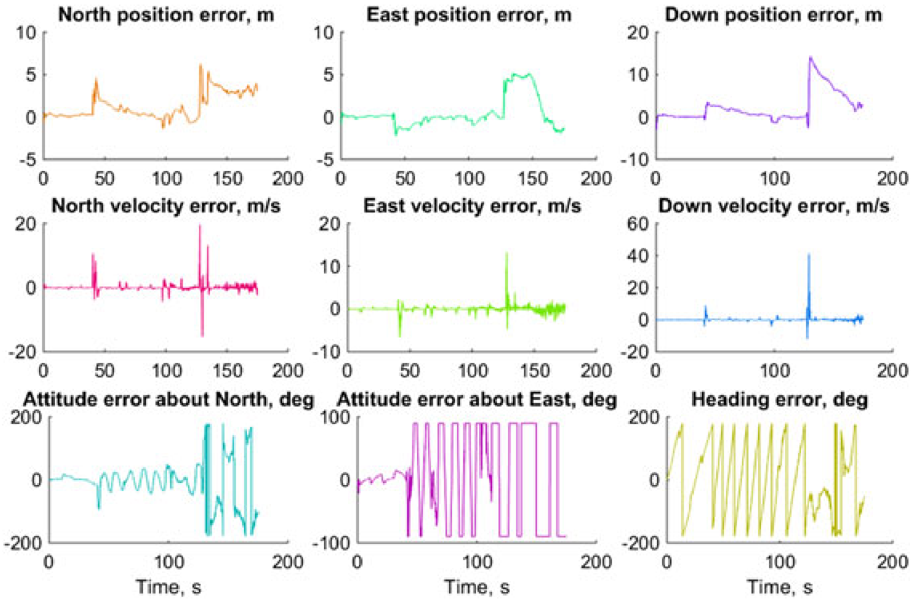


Fig. 2–An example of a typical badly behaved filter's estimation errors in its position attitude and velocity solution. Note how the filter starts well behaved but as soon as the attitude errors about North and East exceed a few tens of degrees (around 30 s) then the variation of all the states becomes extremely erratic.

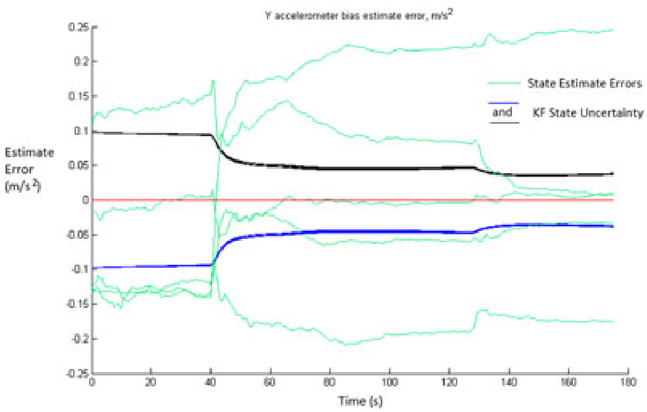


Fig. 4–This shows five runs of a Monte Carlo simulation to illustrate a typical state estimate when a KF integration is behaving inconsistently. In this example some of the estimation errors of this state (Y accelerometer bias) converge to zero but others do not. This spread of estimate errors is not well described by the KF's state uncertainty.

However, when the IMU errors are smaller, the position, attitude, and velocity errors can remain apparently small, within a few degrees and tens of centimeters (per second), even though the IMU error estimates produced are inconsistent with the corresponding state uncertainties (Figure 4). This situation illustrates a problem when determining if a simulation has failed. For a given error distribution, sometimes the IMU errors are within the uncertainty bounds and sometimes they are not, depending on the individual IMU samples. We need the uncertainty to represent the whole distribution not just individual samples. Thus while a particular sensor may happen to have a fortuitously small starting error, with the result that the GNSS integration works well, that is not sufficient. We need to run a large number of different values of the starting errors, sampled from the same distribution, as part of a Monte Carlo simulation, and thus demonstrate that the uncertainty represents the whole distribution.

The intended use of an INS/GNSS integration KF is important when determining failure criteria. Here we wish to calibrate the inertial sensors in the IMU so that the INS can bridge a future GNSS signal outage. This means that we require the errors in KF state estimates corresponding to the IMU's errors, e.g., the accelerometer and gyroscope biases, to end with a smaller distribution than they started. We could make this the test: the simulation must end with a smaller distribution of errors than it started with. However, in the real context some short time later the filter could became unstable and the error estimates go wildly wrong.

Another possible approach could be 'empirical.' We could add a simulated signal outage and consider how well it is bridged by the calibrated solution. However, we can only simulate a short amount of time before the computational cost of running the Monte Carlo simulation becomes too high. For instance, in this paper we only simulate three minutes of motion. If the simulations were continued, the estimates of the states might continue to improve. In some application scenarios, there may be tens of minutes or even hours between outages, so choosing an arbitrary amount of time after which to test whether the resulting calibration is 'adequate' is unsatisfactory. Additionally, determining what is 'adequate' performance for bridging a GNSS outage is not straightforward. For example, two INSs with different amounts of sensor noise would perform differently even if both had their systematic errors perfectly calibrated. Perhaps the GNSS-calibrated INS could be compared to an INS with identical noise parameters but zero systematic errors, although it is not clear how much worse one should expect an imperfectly calibrated INS to behave than a perfectly calibrated one, and still be judged 'adequate.' Therefore, a success criteria based on this approach would be specific to the requirements of a particular application, and thus unsuited to drawing more general conclusions.

In this paper we take a different approach to deciding whether an INS/GNSS integration Kalman filter is 'successful,' we look at filter stability.

In general, a Kalman filter will be stable if all of the assumptions stated in the *INS/GNSS integration based on a Kalman-filter-derivative* section are valid and if the variances and covariances stored in its state error covariance matrix (P-matrix) are an accurate reflection of the real errors of the estimates. Fortunately, if the former condition breaks, that latter tends to also. In a physical system the 'real' errors in these state estimates are not known. However, in simulations, the 'truth' from which the estimates differ is known. This means that stability criteria based on the real estimation errors are possible.

A necessary but not sufficient condition for the state error covariance matrix being correct is that the filter is tuned correctly, as otherwise it is either wrong from the start or will quickly become so.

The stability condition that we test here is the relationship between that filter's state uncertainty for every state at each epoch (averaged across the 100 runs in the Monte Carlo simulation) and the root-mean-squared-error (RMSE) across the set of 100 simulations of the filter's state estimate error. The threshold for failure is that the worst state's RMSE does not exceed 2σ for more than 5% of the simulation time, where σ^2 is the corresponding state variance from the state error covariance matrix. It should be noted that this threshold is very generous, if the state uncertainty, σ, is correct then the RMSE over the 100 Monte Carlo runs should be very close in value to it, as the unbiased estimator of the distribution variance, σ^2, is $\sigma^2 = \frac{n}{n-1}s^2$ where s^2 is the sample variance and n is the number of samples. This means that the probability that this condition would fail by chance is very small, because for a normal Gaussian distribution $P(s > 2\sigma \mid n = 100) \approx 3.8 \times 10^{-37}$. Thus, we can be confident that the state variance is not accurately describing the real errors if this condition fails.

SIMULATION APPROACH

In this section, the approach taken to run simulations to determine whether a particular IMU model is suitable for 'in-run' calibration is outlined. First, the general approach is outlined (*Simulation Philosophy* section), then the particular Kalman filter variant used is discussed (*Basic Kalman Filter* section). This is followed by details of the simulation algorithm's design (*Algorithm Process* section), the filter tuning (*Filter Tuning* section), the motion scenario used (*Simulation Motion Scenario* section), and the GNSS parameters (*GNSS parameters* section). Finally, the way in which the set of inputs were searched is outlined in the *Determining the Search Space* section.

Simulation Philosophy

IMUs are modeled by adding errors to the output of a fictional 'perfect IMU' of differing magnitudes to simulate IMUs of different grades. The values for each of the systematic errors are selected randomly for each run of a Monte Carlo simulation and kept constant within that run. Thus the only input needed to generate the set of systematic errors for all the runs of a Monte Carlo simulation is the standard deviations of each systematic error. Noise sources are specified by their power-spectral-densities (PSDs) and assumed to be white.

IMU sensor noise is generated independently for each run of every test. Similarly, the GNSS measurement errors for each run of each Monte Carlo simulation are selected independently from a common distribution.

Basic Kalman Filter

For the simulations discussed here, the idea was to use the most basic Kalman filter that might realistically be possible. The idea of being to assess the limits of this basic filter and then determine when more complex filters is required. This means that it is both the most simple to program and has the lowest processing load.

The INS/GNSS integration Kalman filter is loosely coupled, which means that the GNSS information is given to the filter in the form of GNSS position and velocity measurements, rather than as, for example, GNSS-pseudoranges. This has the advantage that the results are applicable to other sources of aiding. It is also a standard Kalman filter rather than an extended Kalman filter (EKF) [19, 21] or unscented Kalman filter (UKF) [6]. This means that it has linear system and measurement models. However, it has closed-loop correction of the inertial sensor error states, which improves the stability of the filter if the magnitude of these states was to become large. This is equivalent to an EKF system propagation [1].

The states modeled by this 'basic Kalman filter' are the minimum commonly used configuration, that is, position, attitude, and velocity (3 × 3), accelerometer bias (3) and gyroscope bias (3), a total of 15 states. Potentially better performance could be achieved by estimating the first-order IMU errors as well. However, an additional 27 states would be required to estimate all components of the accelerometer and gyroscope scale factor and cross-coupling errors and the gyroscope g-dependent errors. Modeling additional states makes the filter significantly slower as the majority of the simulation's time is spent performing matrix multiplications, many of which require computational power proportional to n^3, where n is the number of states.

The KF is adapted from the example software provided as open-source on the CD accompanying [1]. The transition, system noise, and measurement matrices used can be found in Equations (14.50), (14.82), and (14.115) of [1], respectively. The calculation is performed in an ECEF reference frame; the inertial sensors have an output rate of 100 Hz, and GNSS (measurement) updates occur at 2 Hz.

Algorithm Process

The process through which the program runs to make a Monte Carlo simulation of a single error distribution is shown in Figure 5. First, the inputs listed in Table 1 are specified, noting that the reason these particular ranges were chosen is discussed in

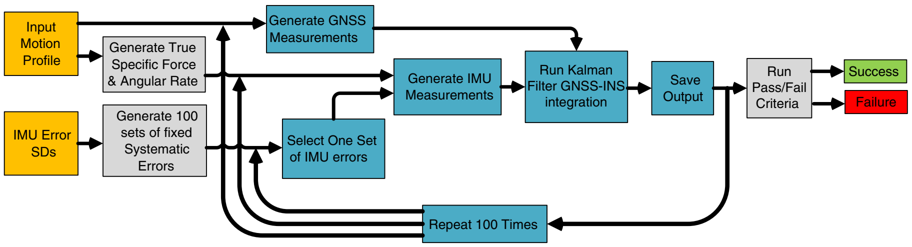

Input Motion Profile
Generate True Specific Force & Angular Rate
Generate GNSS Measurements
Generate IMU Measurements
Run Kalman Filter GNSS-INS integration
Save Output
Run Pass/Fail Criteria
Success
Failure
IMU Error SDs
Generate 100 sets of fixed Systematic Errors
Select One Set of IMU errors
Repeat 100 Times

Fig. 5–Flow Diagram of the Monte Carlo simulation algorithm.

Table 1 — The inputs to the Monte Carlo simulation and their values

Parameter	Value or range used
Number of runs in MC simulation	100
Attitude initialization error SD	0.5 deg (all axes)
Accel. bias SD	1000 to 100,000 μ g
Accel. noise root-PSD	100 to 500 $\mu g/\sqrt{Hz}$
Accel. scale factor error SD	0.06% to 3%
Accel. cross-axis sensitivity SD	0.025% to 1%
Accel. quantization level	$0.01 m/s^2$
Gyro. bias SD	10 deg/h to 20 deg/s
Gyro. noise root-PSD	0.01 to 1.8 deg/$\sqrt{hour}$
Gyro. scale factor error SD	0.03% to 3%
Gyro. cross-axis sensitivity SD	0.02% to 2%
Gyro. g-dependent error SD	1 to 100 deg/hour/g
Gyro. quantization level	0.0002 rad/s

the *Determining the Search Space* section. The true motion profile is also an input, as discussed in the *Simulation Motion Scenario* section.

Next, values for the systematic IMU errors for each of the 100 simulation runs are chosen randomly from a zero-mean Gaussian distribution with standard deviations set to the parameter input values. For example, 100 sets of accelerometer biases are chosen as well as 100 3 × 3 gyroscope g-dependent error matrices. Then the Kalman filter tuning parameters are set using the input distribution SDs, in the manner described in the *Filter Tuning* section.

The next step is to calculate the true specific force and angular rate measurements from the true motion profile, then use all the systematic and stochastic errors to create simulated IMU outputs, and also create simulated GNSS positions and velocities from the true motion profile (*GNSS parameters* section). Then inertial navigation equations and the basic Kalman filter INS/GNSS integration (discussed in the *Basic Kalman Filter* section) are run for each of the 100 sets of simulated IMU and GNSS measurements. The results of all of the simulations are saved.

Finally, summary statistics are calculated for all 100 simulation runs of that distribution, and comparisons between the KF's estimation errors and the uncertainty that it calculates for each state, as discussed in the *Kalman Filter Failure* section, are made. This gives the result of 'pass' or 'fail' for the particular error distribution.

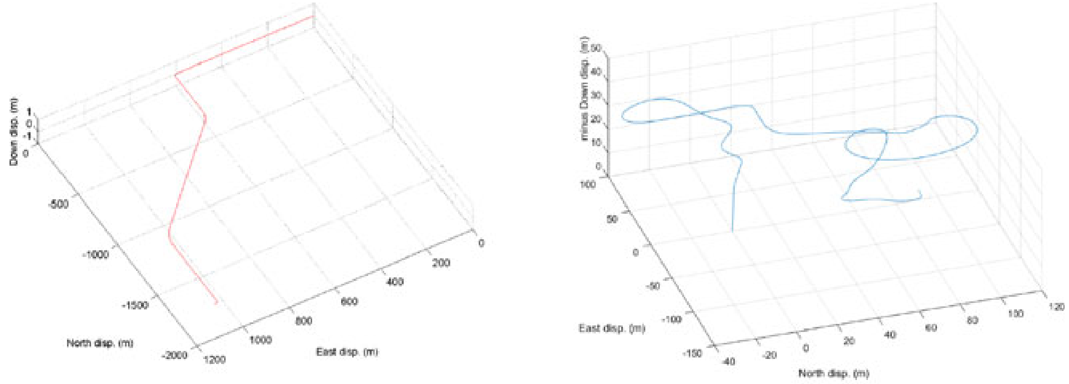
Simulation Motion Scenario

In the research presented in this paper, we chose to use two truth motion profiles for the Monte Carlo simulations.

The first consists of a typical car motion lasting 3 min and containing three turns. It is shown in Figure 6a. This motion profile is fairly representative of the navigation scenario in which a typical consumer-grade IMU might be used. However, as we do not wish to make the results specific to cars or land vehicles we do not implement land-vehicle motion constraints [22], though these would reduce the INS drift and could make the calibration of the IMU systematic errors easier. Additionally, the relatively limited number of different maneuvers makes it relatively difficult to separately observe the different IMU errors in this scenario.

The second motion scenario tested simulates a quadcopter. This was chosen to provide a contrasting example with much higher dynamics. We could also have used motion representative of a fighter aircraft or a missile; however, neither of these are realistic applications for low-cost IMUs. As such, motion was generated representing a small quadcopter of the type available to hobbyists for a few hundred dollars, to carry small payloads such as video cameras.

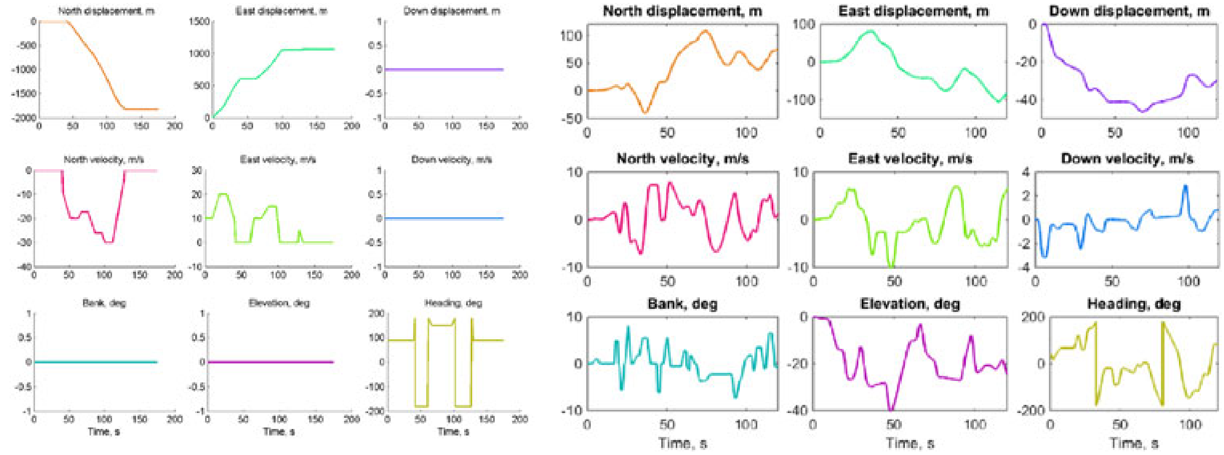
The quadcopter simulated has a maximum speed of 10 m/s, an elevation angle dependent on its speed, and achieves high bank angles when turning.



(a) Car

(b) Quadcopter

Fig. 6–3D projection of the two truth motion profiles used for the Monte Carlo simulations.



(a) Car

(b) Quadcopter

Fig. 7–Components of the two simulated truth motion profiles used to for the Monte Carlo simulations.

These are the advertised performance characteristics of a market-leading model, the DJI Phantom [23]. The 2 min of 3D motion and resulting position, velocity, and attitude are illustrated in Figures 6b and 7b, respectively. Note the frequent sharp turns and that the elevation angle depends on the speed, because the rotors do not pivot with respect to the quadcopter body.

A pedestrian motion scenario has not been tested because IMUs are typically used for pedestrian dead reckoning (PDR) using step detection, which does not require sensor calibration [24, 25]. Inertial navigation using consumer-grade sensors is generally only viable using a foot-mounted IMU, for which zero-velocity updates (ZVUs) are used to calibrate the IMU, requiring a different KF design. Representative pedestrian motion is also difficult to generate [26].

Filter Tuning

As we know the true distribution of the sensor errors, we could tune the KF with precisely these values. However, as is standard practice, the error distributions and noise power spectral densities are over-bounded in the KF integration [1]. This helps maintain filter stability, as those error sources that are not estimated and the effects of any nonlinearities can appear as noise to the filter. The noise root-PSD assumed within the KFs is set to double the noise root-PSD used in the inertial sensor simulations, plus an additional factor detailed below. Additionally, the initial position, velocity, and attitude states uncertainties are overmodeled by a factor of two.

The sensor bias states could be tuned using only the stochastic parameters of the biases themselves (as we initially assumed in [14]). However, we observed that particular state estimates were frequently becoming unstable due to systematic errors that a basic 15-state Kalman filter does not directly model, such as scale factor and cross-coupling errors. Thus, we decided to take these unestimated higher-order errors into account when tuning the filter.

The basic Kalman filter (*Basic Kalman Filter* section) assumes that all the sensor errors are a combination of bias and noise, so any other error types will manifest either in the estimates of bias or as additional 'noise.' The bias-like effects of the higher-order errors are the average effect that these have on the measurements of specific force and angular rate. This is equivalent to the effect of the higher-order errors on the average specific force and angular rate applied in the simulation.

The angular rates and horizontal specific forces (accelerations) roughly average to zero across the whole simulation, while the vertical (z) specific force averages to about 9.8 ms^{-2} due to reaction to Earth's gravity. Consider the scale factor and cross-coupling error matrix, $\mathbf{M_a}$, where $s_{a,i}$ is the scale factor error of the i-axis sensor and $m_{a,ij}$ is the cross-coupling error on the i-axis sensor of the j-axis specific force, so

$$\mathbf{M_a} = \begin{pmatrix} s_{a,x} & m_{a,xy} & m_{a,xz} \\ m_{a,yx} & s_{a,y} & m_{a,yz} \\ m_{a,zx} & m_{a,zy} & s_{a,z} \end{pmatrix}. \tag{1}$$

The entries in the third column are those depending on the vertical-axis specific force. As a result, these errors appear (on average) to the simplified IMU model used in the basic Kalman filter (*Basic Kalman Filter* section) as an additional bias and noise on the accelerometer signals, whereas the other six errors only appear as noise. In the equivalent M-matrix for the gyroscope, all nine errors must be absorbed by the system noise. However, for the gyroscope g-dependent error matrix, which is a 3 × 3 matrix describing the effect of the specific force on the angular rate measurements, the three components corresponding to the z-axis specific force appear to produce an additional bias, and all components appear to produce additional gyroscope noise.

In order to account for the bias in the tuning, the state uncertainties associated with the accelerometer biases are set equal to the root sum of squares (RSS) of the accelerometer bias SD and g times the greater of the scale factor and cross-coupling error SDs. We chose to use this form because it allows arbitrary orientation of the IMU. Similarly, the gyroscope bias uncertainties were set to the RSS of the gyroscope bias SD and the SD of the g-dependent bias associated with a specific force of g.

In order to maintain filter stability, it is necessary to include additional system noise (represented by the 'Q'-matrix) to account for the noise-like effect of the higher-order systematic errors. The best way to tune the KF noise parameters to account for these errors depends both on the dynamics of the motion and the magnitude of the unestimated errors. These error sources add a large amount of 'noise' to the sensor outputs during a maneuver and have little impact the rest of the time. In a real system it is likely that the designer would determine the amount of 'extra noise' empirically, using trial-and-error until the system remains stable. Some authors also suggest using a system model that depends on the dynamics (see [1]). Our intention is to test the most basic KF integration possible so we have added a constant amount of additional system noise noise covariance that is proportional to the variance of the unestimated systematic errors.

The system noise used in the KF to represent the behavior of the accelerometers is given by,

$$S_a^{\mathrm{KF}} = 2^2 S_a^{\mathrm{IMU}} + \left(\max\left(\sigma_{m_a}, \sigma_{s_a}\right) k_f\right)^2 t_a, \quad (2)$$

where S_a^{KF} is the accelerometer noise PSD modeled within the KF, S_a^{IMU} is the accelerometer noise PSD used to generate the IMU model, k_f is a constant of proportionality to account for the level of variation in the specific force during the motion and, σ_{m_a} and σ_{s_a} are SDs of the accelerometer cross-axis sensitivity and scale factor error used to generate the IMU model, respectively. t_a is the smoothing time described below.

Similarly, the system noise used in the KF to represent the behavior of the gyroscopes is given by,

$$S_g^{\mathrm{KF}} = 2^2 S_g^{\mathrm{IMU}} + \left(\max\left(\sigma_{m_g}, \sigma_{s_g}\right) k_\omega\right)^2 t_a + \left(\sigma_{G_g} k_f\right)^2 t_a, \quad (3)$$

where S_g^{KF} is the gyroscope noise PSD modeled within the KF, S_g^{IMU} is the gyroscope noise PSD used to generate the IMU model, k_ω is a constant of proportionality to account for the level of variation in the angular rate during the motion and, σ_{m_g}, σ_{s_g}, and σ_{G_g} are SDs of the gyroscope cross-axis sensitivity, scale factor error, and g-dependent error used to generate the IMU model, respectively.

The constants of proportionality, k_f and k_ω are calculated based on statistical properties of the true motion. These two constants have the same units as specific force and angular rate, respectively. They are the average absolute difference in specific force or angular rate smoothed over 10 GNSS epochs and over 1 epoch for each motion profile. That is,

$$\begin{aligned} k_f &= \mathrm{mean}\left(\left|\mathrm{M}\left(\mathbf{f}_{\mathbf{ib}}^{\mathbf{b}}, t_u\right) - \mathrm{M}\left(\mathbf{f}_{\mathbf{ib}}^{\mathbf{b}}, 10t_u\right)\right|\right) \\ &= \begin{cases} 0.1058\ \mathrm{ms}^{-2}, & \text{for car motion} \\ 0.2302\ \mathrm{ms}^{-2}, & \text{for quadcopter motion} \end{cases} \end{aligned} \quad (4)$$

where $\mathbf{f}_{\mathbf{ib}}^{\mathbf{b}}$ is the specific force and t_u is the KF measurement update time (0.5 s). The function "$\mathrm{M}(a,t)$" takes a moving average of the time series a over t seconds. "| |" is the absolute value operator.

$$\begin{aligned} k_\omega &= \mathrm{mean}\left(\left|\mathrm{M}\left(\boldsymbol{\omega}_{\mathbf{ib}}^{\mathbf{b}}, t_u\right) - \mathrm{M}\left(\boldsymbol{\omega}_{\mathbf{ib}}^{\mathbf{b}}, 10t_u\right)\right|\right) \\ &= \begin{cases} 0.008228\ \mathrm{rad/s}, & \text{for car motion} \\ 0.042038\ \mathrm{rad/s}, & \text{for quadcopter motion} \end{cases} \end{aligned} \quad (5)$$

where $\boldsymbol{\omega}_{\mathbf{ib}}^{\mathbf{b}}$ is the angular rate. The time constant for Equations (2) and (3) is 10 times the GNSS (measurement) update rate ($t_a = 10t_u$), as this is the period of the moving average.

GNSS Parameters

The GNSS simulation settings are chosen to simulate the operation of consumer-grade GNSS user equipment in a relatively benign signal environment. This is reasonable as one would only try to use GNSS to calibrate the INS when good reception is available.

As the focus of this paper is on the IMU calibration, we have used a relatively crude GNSS model. Bias-like GNSS errors were neglected as they only affect position determination while we are aiming to assess the INS calibration. Additionally, as the simulation is relatively short (3-minutes) the time variation in GNSS systematic errors has been neglected. A constellation of 30 satellites was simulated. The effect of code tracking error is simulated by white noise with a SD of 1 m and 0.02 m/s on the pseudo-range and pseudo-range-rate, respectively. Note that the KF update-interval exceeds the correlation time of typical GNSS tracking errors.

The measurements used for the Kalman filter are GNSS-like generic position and velocity measurements, i.e., loosely-coupled integration. The associated measurement noise SD KF tuning parameters are 2.5 m and 0.1 m/s on each axis for the position and velocity, respectively.

Determining the Search Space

There are a large number of possible variables that could be changed to run different Monte Carlo simulations, which were listed in Table 1. In order to be able to run enough simulations using limited computational resources the Monte-Carlo simulations contain 100 runs each.

Given there are a vast number of possible combinations of the 12 possible input parameters, we made some assumptions to reduce the number of potential combinations.

First, only levels of error that exist for real sensors from tactical to consumer MEMS grade were tested. Thus, there were no 'zero-bias' or 'zero-noise' tests, nor were any sensors simulated with unrealistically large errors. The ranges for each parameter chosen are given in Table 1. Some real sensor specifications are provided in Table 2 to demonstrate that the ranges tested encompass currently available sensors.

Second, we have chosen to vary some but not all of the possible inputs. For instance, we are fixing the sensor quantization and the attitude initialization error. Also, it seems very unlikely that certain combinations of errors exist, such as high cross-coupling with extremely accurate scale factor. For this reason, we vary some of the errors together, with one parameter for the higher-order accelerometer 'unestimated' errors encompassing accelerometer scale factor error and accelerometer cross-coupling, and a second parameter for the higher-order gyroscope 'unestimated' errors, which combines gyroscope scale factor error, gyroscope cross-coupling, and gyroscope g-dependent error. These two error parameter sets are split into five levels for testing: tactical, low, medium, high and very high (See Table 3).

This leaves six different distribution parameters to test: accelerometer bias; gyroscope bias; accelerometer noise; gyroscope noise; and the two unestimated error parameters. As we are trying to find the border between 'success' and "failure," we are looking for a five-dimensional subspace in six-dimensional space, analogous with how a surface is a two-dimensional subspace of three-dimensional space.

First, we determine which are the most interesting areas of the search space by testing points over a coarse grid (7 (accel bias) ×9 (gyro bias) ×5 (accel noise) ×5 (gyro noise) ×5 (higher-order accel) ×5 (higher-order gyro)). Then having identified the general structure of the space, border regions were re-searched on a much finer grid to find the 'edge,' using a strategy that tests along one parameter until adjacent points are found where one is a pass and the other a fail, then incrementing a second parameter and returning to varying the first. This finer grid has a geometrical spacing where a point is 110% of

Table 3 — The five levels of higher-order unestimated IMU errors tested

Parameter std. dev.	'tactical'	'low'	'medium'	'high'	'very high'
Accelerometer					
Scale factor error	0.06%	0.5%	1%	2%	3%
Cross-axis sensitivity	0.025%	0.25%	0.5%	0.75%	1%
Gyroscope					
Scale factor error	0.03%	0.3%	1%	2%	3%
Cross-axis sensitivity	0.02%	0.2%	1%	1.5%	2%
g-dependent error (deg/hour/g)	1	5	10	50	100

Table 2 — A selection of sensor error distributions derived from their datasheets, in the units given. The Bosch, ST, and Invensense are consumer-grade MEMS. The Xsens is a factory-calibrated MEMS IMU, which costs around $2,500. Where a range is given this parameter depends on the full-scale measurement range selected

Sensor manufacturer	Bosch	STMicroelectronics	Invensense	Xsens
Model	BMA180 [27]	L3G4200D [28]	MPU-9150 [29]	MTi-G [30]
Type	accelerometer	gyroscope	single-chip IMU	factory-calibrated IMU
Accelerometer Errors				
Bias	± 0.588 m/s^2	n/a	± 0.784(x&y) to 1.47(z) m/s^2	0.02 m/s^2
Noise	0.00147 $m/s^2/\sqrt{Hz}$	n/a	0.0039 $m/s^2/\sqrt{Hz}$	0.002 to 0.004 $m/s^2/\sqrt{Hz}$
Scale factor error	±1.5% to 3%	n/a	± 3%	± 0.03%
Cross-axis sensitivity	1.75%	n/a	not specified	aligned to ±0.1 degree
Nonlinearity	0.15 to 0.75 % FS	n/a	0.5 % FS	not specified
Gyroscope errors				
Bias	n/a	± 10 to 75 deg/s	±20 deg/s	±1 deg/s
Noise	n/a	0.03 deg/s/$\sqrt{Hz}$	0.005 deg/s/$\sqrt{Hz}$	0.05 to 0.1 deg/s/$\sqrt{Hz}$
Scale factor error	n/a	±4% [31]	± 3%	not calibrated
Cross-axis sensitivity	n/a	not specified	± 2%	aligned to ±0.1 degree
Nonlinearity	n/a	0.2% of FS	0.2 % FS	not specified

the value of the previous point, and thus, it appears equally spaced on a log scale. Thus, each graph in the *Results* section has more data points along the pass-fail boundary than elsewhere.

RESULTS

Discussion of the results of the simulations is split into three parts. The first two parts concern the road vehicle simulation: *Road vehicle results when the higher-order IMU error parameters are 'small'* section presents results with very small higher-order error parameters (scale factor, cross-coupling, and gyro g-dependent error); *The effect of the higher-order IMU error parameters* section presents results with higher-order errors large enough to impact performance. *Quadcopter Results* section examines the results of the quadcopter simulations. *Discussion* section discusses these results.

Road Vehicle Results when the Higher-Order IMU Error Parameters are 'Small'

When the higher-order IMU error parameters are very small, the assumptions we made by using a Kalman filter (KF) with only 15 states are reasonable. That is, in relation to these errors, the filter should be fit for purpose and, if the filter 'breaks,' it must be for a different reason. However, there are error distributions for which the KF fails even with 'tactical' unestimated errors. These are illustrated in Figures 8 and 9. As it is impractical to the present six-dimensional diagrams, we present three-dimensional diagrams that show the variations in three parameters with the other three parameters fixed.

Observing the border between 'success' and 'failure' in these figures, it is clear that the most important error parameter is gyroscope bias. All of the distributions tested with a gyroscope bias standard deviation (SD) below 0.75 degrees per second (deg/s), equivalent to 2705 degrees per hour (deg/h), were successful, and all those tested above 2.6 deg/s (9400 deg/h) were failures. Note that this failure point is between the specified performance of factory-calibrated IMUs and consumer-grade MEMS gyroscopes (Table 2). This makes it clear that the accuracy of the attitude solution is key to INS/GNSS KF stability. This is primarily due to the use of the small angle approximation (*INS/GNSS integration based on a Kalman-filter-derivative* section).

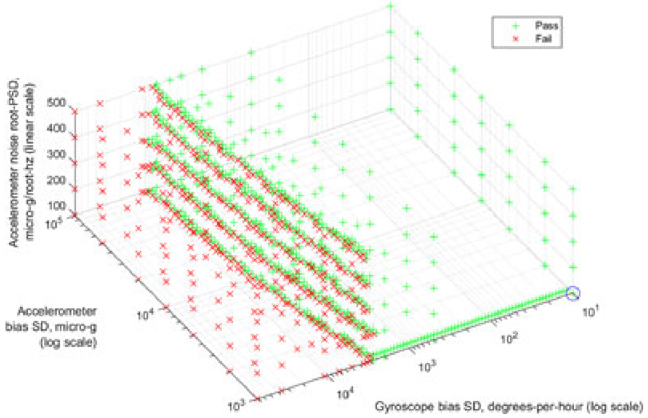


Fig. 8–Three-dimensional subspace of the search space showing where the KF fails with road vehicle motion for different values of accelerometer and gyroscope bias and accelerometer noise with gyroscope noise fixed at 0.01 deg per root hour and both accelerometer and gyroscope unestimated errors fixed to 'tactical' (see Table 3).

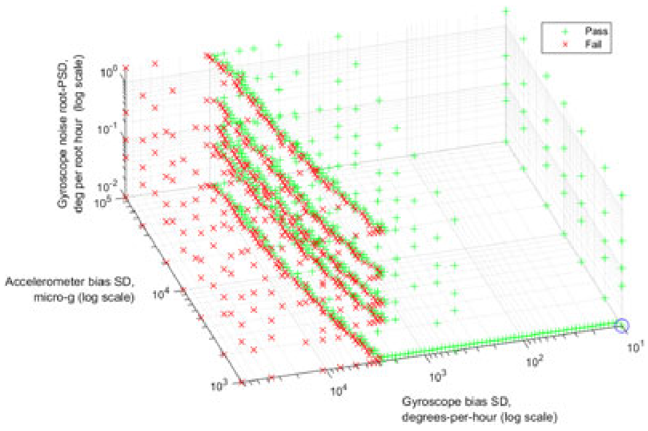


Fig. 9–Three-dimensional subspace of the search space showing where the KF fails with road vehicle motion for different values of accelerometer and gyroscope bias and gyroscope noise with accelerometer noise fixed at 100 $\mu g/\sqrt{Hz}$ and both accelerometer and gyroscope unestimated errors fixed to 'tactical' (see Table 3).

Variation of the 'border' between success and failure according to the other IMU error parameters can also be observed. The accelerometer bias is most significant with the maximum tolerable gyroscope bias SD ranging from 2705 deg/h for a 1 milli-g accelerometer bias SD to 9337 deg/h for a 100 milli-g accelerometer bias SD. It is perhaps surprising that worse accelerometer bias performance allows the filter to cope with more gyroscope bias. A possible explanation is that the larger accelerometer bias uncertainty in the Kalman filter indirectly results in larger attitude uncertainties, enabling it to tolerate greater divergence from the small angle approximation.

The effect of the IMU's two noise parameters on the position of the boundary is even smaller. The accelerometer noise affects the maximum tolerable gyroscope bias SD by a factor of about 1.25 (Figure 8), while the effect of the gyroscope noise is less than a factor of 1.1 over the range simulated (Figure 9).

The Effect of the Higher-Order IMU Error Parameters

When we vary the parameters which are not modeled as states in the Kalman filter, we know that the KF is only suitable if these parameters are insignificant. Thus, the question we are asking is: What size of error is insignificant?

As described in the *Determining the Search Space* section, we tested five levels of higher-order errors of the gyroscopes and accelerometers, as shown in Table 3. The unestimated errors gyroscope and accelerometer were varied separately. Both of these parameters were set to 'tactical' for the results discussed in *Road vehicle results when the higher-order IMU error pa- rameters are 'small'* section. Here, we present the results when one or both of the gyroscopes and accelerometers higher-order error parameters were set to low, medium, high, or very high. Note that the results presented here show a considerable improvement from the preliminary results presented in [14] because of the improved filter tuning, presented in the *Filter Tuning* section.

First, we examine the effect of increasing the gyroscope and accelerometer higher-order error parameters individually. Figure 10 shows the effect of increasing the higher-order accelerometer error parameters through each of the five levels while holding the two noise parameters and the gyroscope higher-order errors at 'tactical' level. As the level of unestimated error increases (vertical axis), a new unstable area forms and grows in the low-bias corner of the graph, which combines with the pass/fail boundary discussed in the *Road vehicle results when the higher-order IMU error parameters are 'small'* section, to only leave a small stable area when the higher-order errors are very high. Figure 11 is the analogue showing the effect of the gyroscope unestimated error parameter. A similar effect can be observed. However, the unstable area grows more for each level of unestimated error, leaving only an extremely small pass area in the high-bias corner when the unestimated error parameter reaches high and no passes at all for very high higher-order errors.

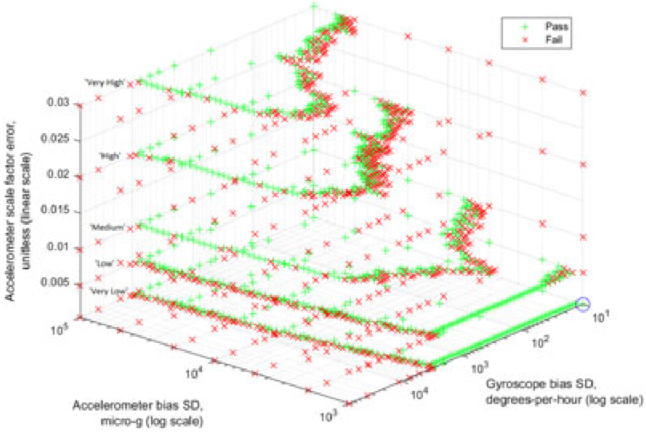


Fig. 10–Three-dimensional subspace of the search space showing where the KF fails with road vehicle motion for different values of accelerometer and gyroscope bias and accelerometer unestimated errors with the other errors fixed to 'tactical' level (see Table 3).

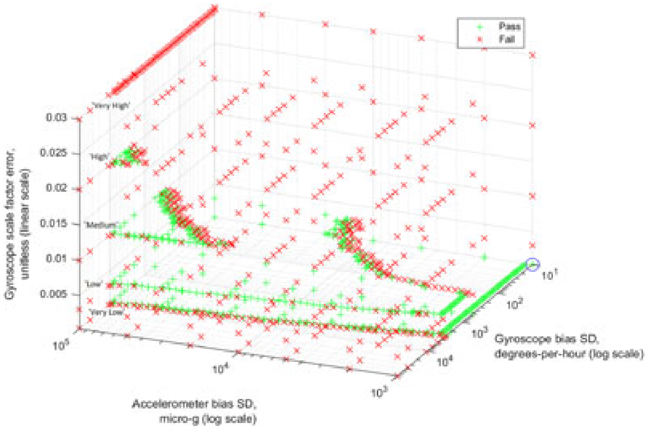


Fig. 11–Three-dimensional subspace of the search space showing where the KF fails with road vehicle motion for different values of accelerometer and gyroscope bias and gyroscope unestimated errors with the other errors fixed to 'tactical' level (see Table 3).

Quadcopter Results

When the higher-order errors are at a low enough level to be insignificant, e.g., at 'Tactical' level, the filter stability is similar to that for the car motion. In Figure 12, which represents filter behavior for insignificant levels of unestimated error, the behavior is qualitatively almost exactly the same as the equivalent figure for car motion, Figure 8. It exhibits a maximum gyroscope bias SD beyond which the filter becomes unstable with some dependence on the accelerometer bias SD, but minimal variation with the accelerometer and gyroscope noise levels. Numerically, maximum possible gyroscope bias SD has increased slightly, ranging from 3684 deg/h for a 1 milli-g accelerometer bias SD to 12430 deg/h

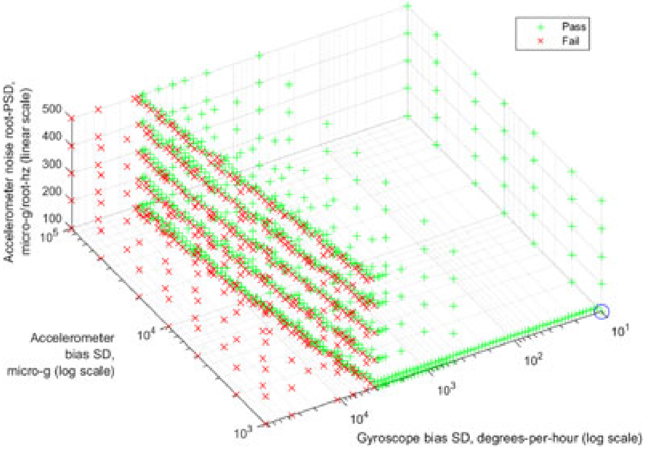


Fig. 12–Three-dimensional subspace of the search space showing where the KF fails with quadcopter motion for different values of accelerometer and gyroscope bias and accelerometer noise with gyroscope noise fixed at 0.01 deg per root hour and both accelerometer and gyroscope unestimated errors fixed to 'tactical' (see Table 3).

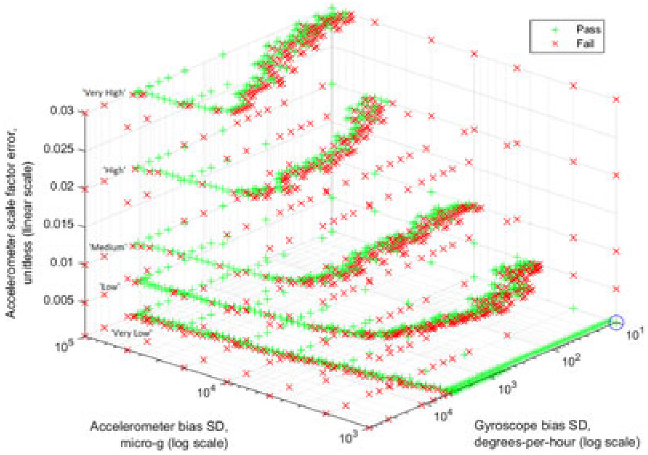


Fig. 13–Three-dimensional subspace of the search space showing where the KF fails with quadcopter motion for different values of accelerometer and gyroscope bias and accelerometer unestimated errors with other errors fixed at 'tactical' level (see Table 3).

for a 100 milli-g accelerometer bias SD. This slight movement could be due to the biases becoming more observable with the different maneuvers.

As the level of higher-order errors increase, the differences in filter performance between the car and quadcopter motion profiles become larger. Figure 13 illustrates the changes in the stable area with different levels of accelerometer higher-order errors; it is the quadcopter analog of Figure 10. Therefore, the maximum gyroscope bias uncertainty boundary is the same as in Figure 12, just as the boundary in Figure 8 was repeated in Figure 10. However, the range of accelerometer biases that are stable with lower gyroscope biases are considerably smaller with the quadcopter than the equivalent with car motion. When we compare non-negligible levels of accelerometer higher-order error, between Figures 10 and 13, the minimum acceptable accelerometer bias has reduced significantly, particularly around 100 deg/hr gyroscope bias.

Figure 14 shows the effect of different levels of gyroscope higher-order errors. The analogous behavior for car motion is shown in Figure 11. With very high levels of unestimated errors, neither motion profiles enable a stable KF. At high and medium levels of the higher-order gyroscope errors, the stability regions are very small for the car motion and smaller still for the quadcopter motion. Thus, it is very difficult to achieve stability by exaggerating the bias SDs assumed within the KF. When the unestimated errors are low, the high-gyro-bias side of the stability area is very similar; however, on the low-gyroscope-bias side, the quadcopter motion is only stable for higher levels of bias. Thus, the assumed gyroscope bias would need to be exaggerated to achieve stability.

These results indicate that higher dynamics have a marked additional de-stabilizing effect on the filter and this manifests as the stable area shrinking significantly, pushing it further towards the high-bias corner. The impact of these errors on filter stability with quadcopter motion is approximately equivalent to the impact of unestimated errors half a level higher than on car motion.

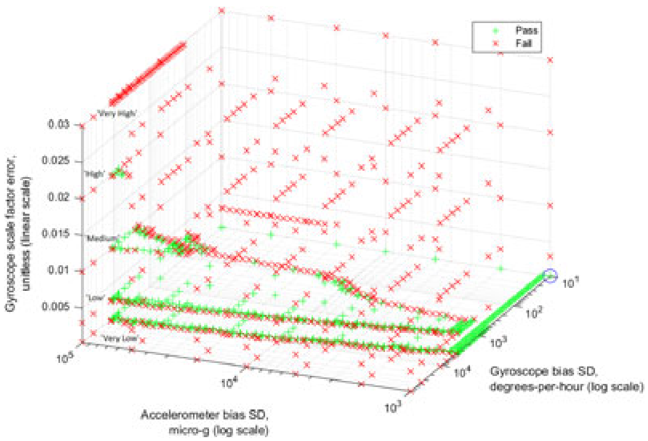


Fig. 14–Three-dimensional subspace of the search space showing where the KF fails with quadcopter motion for different values of accelerometer and gyroscope bias and gyroscope unestimated errors with other errors fixed at 'tactical' level (see Table 3).

Discussion

Road vehicle results when the higher-order IMU error parameters are 'small' section showed that when the levels of the unestimated higher-order errors were very low, the noise level for both gyroscope and accelerometer noise made only a small difference to the maximum tolerable bias SDs. However, when the higher-order errors are larger, higher sensor noise aids filter stability. This can be seen by comparing Figures 13 and 15, noting the significant differences in the 'low-accelerometer bias SD' boundary that comes about when the accelerometer and gyroscope noise parameters are at the high rather than the low end of the range tested.

We aim to explain this and the other behavior observed in the previous sections by considering two things. First, consider how the unestimated error sources appear to the filter. As discussed in the *Filter Tuning* section, their average effect appears as bias and the remainder must be treated as noise, which is not white, but correlated over each maneuver. The tuning of the initial state uncertainties was adjusted to account for the bias-like components. However, this compensation cannot be totally perfect, and the higher the unestimated errors, the more significant this is. Second, while the KF tuning adds additional system noise to account for the noise-like behavior, it adds it constantly throughout the simulation, as we are not using an adaptive KF. This means that most of the time, we are adding more system noise than necessary, while during maneuvers, when the higher-order errors are excited, we are not always adding enough.

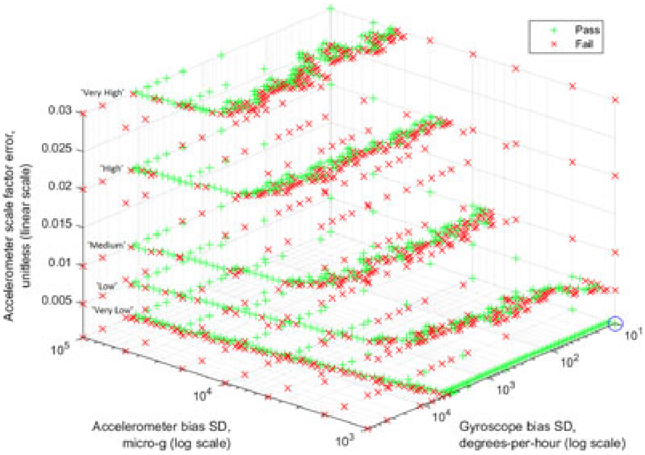


Fig. 15–Three-dimensional subspace of the search space showing where the KF fails with quadcopter motion for different values of accelerometer and gyroscope bias. The gyroscope noise is fixed at 1.8 deg per root hour, the accelerometer noise fixed at 500 $\mu g/\sqrt{Hz}$ and accelerometer unestimated errors with gyroscope unestimated errors fixed at 'tactical' level (see Table 3). It is the equivalent of Figure 13 with more noise.

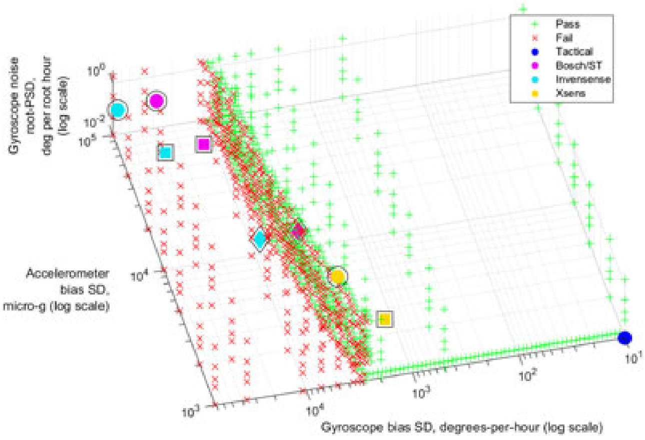


Fig. 16–Three-dimensional subspace of the search space showing where the KF fails for different values of accelerometer and gyroscope bias and gyroscope noise with accelerometer noise fixed at 100 $\mu g/\sqrt{Hz}$ and both accelerometer and gyroscope unestimated errors fixed to very low (see Table 3). Also shown are the specifications of selected MEMS sensors: a tactical grade IMU (dark blue), Xsens Mti-G (yellow), Invensense MPU-9150 (light blue), and an IMU comprising a STMicrotronics L3G4200D gyroscope and a Bosch BMA-180 accelerometer (magenta), see legend. The circles denote the performance of a single sensor, the squares denote the theoretical performance of a small array of four sensors, and the diamonds a large array of 64 sensors (not plotted for Xsens).

In order to absorb the unestimated IMU error parameters without modeling them as states, there needs to be both enough overmodeling of the bias states to cover the extra bias from the higher-order errors and enough overmodeling of the noise to cover the extra noise. The filter tuning used here to account for biases and noise takes a constant multiple of the actual noise and bias to configure the KF for the system noise and bias state uncertainty (respectively). Therefore, when the noise or bias is higher, the overmodeling is also higher in absolute terms, as it is fixed in proportion to the noise and bias. This would explain why the filter can cope with higher unestimated errors when the biases or noise are higher.

This and the slightly counter-intuitive results discussed in the *Road vehicle results when the higher-order IMU error parameters are 'small'* section lead to the observation that, in cases where a filter is close to the stability limits, it may be possible to aid KF stability by exaggerating the assumed accelerometer bias uncertainty when the unestimated error parameters are larger.

The differences between the motion profiles have already been discussed in the *Quadcopter Results* section.

PRACTICAL APPLICATIONS OF THE RESULTS

Basic Application

One of the aims of this research is to enable navigation system designers to look up the specifications of the IMU they are planning to use and determine whether a basic Kalman filter INS/GNSS integration will be sufficient to calibrate the sensors in-run. This would save them the time and effort of designing and building the whole system only to find that the sensors' errors are too large for a standard KF to remain stable. In practice, there will be some sensors for which the KF can be stabilized by exaggerating some of the errors specified within the KF's tuning parameters.

As an example of how this could be applied, the real sensor models, whose specifications were given in Table 2, are plotted in Figure 16 as different colored circles. It is clear that the two uncalibrated automotive MEMS sensors are well outside the KF stability limit; and the Xsens IMU, which is factory-calibrated, is right on the boundary. It is also noteworthy that the specification of the Xsens as "aligned to 0.1 degree" [30] is very close to the 'tactical' higher-order errors tested here. As such, these results suggest that basic KF integration using the Xsens should be stable, provided that the tuning is handled carefully.

Array Techniques

A sensor array is where the outputs of multiple inertial sensor triads (accelerometers, gyroscopes, or both) are combined by ensemble averaging to a single angular rate and specific force output. This can dramatically increase the performance of the combined system by reducing both the noise and the distribution of the systematic errors.

The effects of using a sensor array can be seen in Figure 16. The circles represent single sensors, but an array of four of these would have halve the systematic errors and the noise PSD as illustrated by the squares. And an array of 16 would have a quarter the level of noise and systematic errors assuming zero-mean distributions. There is an upper limit on the number of sensors that can reasonably be combined into an array. The performance of 64 sensors are shown as diamonds as an example of a large, but still practical array (for instance [32] has an array of 100 sensors). It can be seen that the effect of even a small array moves the Xsens sensor well inside the stability boundary but even with the large array of combined Bosch accelerometers and ST gyroscopes has only just made it to the boundary, and the Invensense MPU-9150 array is still far away. This implies that even when using a large array, some pre-calibration of the raw sensors may be required.

If the assumptions above such as zero-mean distributions, Gaussian distributions, and independence between sensor triads are not met, then in a normal sensor array, performance will be reduced. However, a smart array can be used instead. This arranges the constituent inertial sensor triads which have been arranged to exploit the design characteristics of the sensors such that the errors exhibited are smaller than those obtained by simple averaging [13]. For example, if the bias does not have a zero-mean distribution across the constituent sensors of the array, then an anti-parallel arrangement of the sensor triads (with their outputs differenced) will make the combined smart array's bias zero mean [13] as well as removing most of the temperature dependent bias variation [33].

Smart array techniques are important because real sensors' errors are not distributed with perfect zero-mean normal distributions, for examples, see [13] and [31]. There could be a number of reasons why sensors' errors are not distributed independently with zero-mean normal distributions. These could range from 'cherry picking' the sensors with the smallest errors to be sold as higher grade sensors, effectively removing the middle of the distribution, to manufacturing or design limitations. It is also likely that sensors from the same production batch will have errors that are more correlated than others of the same design.

Other Implications

When considering the widest implications of the research presented in this paper, first, we have to consider how it is limited. First, while the two motion scenarios tested represent two likely applications of MEMS IMUs with markedly different levels of dynamics, they do not cover every possible use case.

Second, we have considered the simplest possible INS/GNSS integration Kalman filter possible, as it is currently configured and it is not appropriate for even medium higher-order gyroscope errors, which are relatively small in MEMS terms, although it is possible that further relaxing the filter tuning may regain stability at the cost of convergence speed and accuracy. Estimating the higher-order errors as KF states would help mitigate their effects. Even where these errors are difficult to observe, their inclusion as states provides the KF with a much more realistic system error model. However, it is very unlikely that this would enable toleration of gyroscope biases larger than can be tolerated with 'tactical' level higher-order errors.

Bearing these limits in mind, we can make a couple of important points. The first is that basic KF integration can be sufficient to calibrate both accelerometer and gyroscope biases if the gyroscope bias SD is below the level of around 1–2 deg/s. This means that any pre-calibration should concentrate on the gyroscope bias and estimating the other parameters can be left to the INS/GNSS integration, if the unestimated errors are sufficiently small. Fortunately, gyroscope bias is also among the easiest of the IMU errors to pre-calibrate, as it can be observed in a simple static test.

A major issue for calibration of all MEMS sensors is that the systematic errors vary with temperature. For example, the L3G4200D gyroscope bias variation with temperature is specified at 0.03 deg/s/°C [28]. However, even with a 30°C operating range, this is less than 1 deg/s. Thus, if the sensor were already calibrated before a change in temperature, then simply increasing the state uncertainty in the KF when the temperature changes should allow the INS/GNSS integration to compute the new bias.

CONCLUSIONS

In this paper, a new simulation approach to determine the limit of in-run sensor calibration using a basic Kalman filter integration has been presented. Stability criteria are defined based on the consistency of the state estimation errors and their uncertainties.

Monte Carlo simulations using a basic INS/GNSS Kalman filter to calibrate the sensor biases have been used to determine which IMUs are good enough for in-run calibration. It is important to note that this technique is not specific to GNSS integration, another aiding technology or zero-velocity updates could be used instead.

Both car and UAV motion have been simulated. These tests have shown that within the ranges of errors encountered by real MEMS inertial sensors, the most important error is the gyroscope bias. The maximum gyroscope bias standard deviation that is acceptable varies between 0.75 and 2.6 deg/s depending on the accelerometer noise level, gyroscope noise level, and accelerometer bias SD, the latter being the most important.

A major limitation is the size of the higher-order errors, such as scale factor and cross-coupling errors and the gyroscope g-dependent errors, that are present in typical IMUs. To maintain KF stability, these must be estimated as Kalman filter states, pre-calibrated and/or averaged out using an array technique.

The results of this study enable navigation system designers to determine for which IMU specification levels they need to use a more complex and/or nonlinear Kalman filter variant, and/or pre-calibrate the sensors.

The outputs of this study can also support the prioritization of research into ultra-low-cost inertial navigation between nonlinear estimation, pre-calibration techniques, smart arrays, and hardware development. This will open up many new applications for inertial navigation, including smartphone apps, intelligent transport systems, dismounted soldiers, and autonomous vehicles.

Further research could examine the stability limits of higher-order estimation algorithms such as the unscented Kalman filter.

ACKNOWLEDGEMENTS

Henry Martin is jointly funded by the Engineering and Physical Sciences Research Council (EPSRC) award number 10001940 and BAE Systems Applied Intelligence Laboratories. The authors would like to thank Santosh Bhattari (UCL) for the useful discussions on Kalman filtering and Tuula Eriksson (UCL) for her help with proofreading. The software library to run the simulations in this paper can be obtained by enquiry to p.groves@ucl.ac.uk.

REFERENCES

1. Groves, P. D., *Principles of GNSS, Inertial, and Multi-Sensor Integrated Navigation Systems*, 2nd ed., Boston London: Artech House, 2013.
2. Aggarwal, P., Syed, Z., Noureldin, A., and El-Sheimy, N., *Mems-based Integrated Navigation*, Boston London: Artech House, 2010.
3. Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME, Journal of Basic Engineering*, Vol. 82, Series D, 1960, pp. 35–45.
4. Cox, D. B. Jr., "Integration of GPS with Inertial Navigation Systems," *NAVIGATION*, Vol. 25, No. 2, 1978, pp. 236–245.
5. Farrell, J., *Aided Navigation: GPS with High Rate Sensors*, Series, McGraw-Hill Professional Engineering: Electronic Engineering, New York: Series, McGraw-Hill Education, 2008.
6. Julier, S. J., and Uhlmann, J. K., "A New Extension of the Kalman Filter to Nonlinear Systems," *Proceedings of AeroSense: The 11th International Symposium on Aerospace Defense Sensing, Simulation, and Controls,* SPIE, Orlando, FL, USA, 1997, pp. 182–193.
7. Gustafsson, F. et al., "Particle Filters for Positioning, Navigation and Tracking," *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, 2002, pp. 425–437.
8. Mehra, R. K., "Approaches to Adaptive Filtering," *IEEE Transactions on Automatic Control*, Vol. AC-17, 1972, pp. 693–698.
9. Mohammed, A. H., and Schwartz, K. P., "Adaptive Kalman Filtering for INS/GPS," *Journal of Geodesy*, Vol. 73, 1999, pp. 193–203.
10. Magill, D. T., "Optimal Adaptive Estimation of Sampled Stochastic Processes," *IEEE Transactions on Automatic Control*, Vol. AC-10, 1965, pp. 434–439.
11. Hide, C., Moore, T., and Smith, M., "Adaptive Kalman Filtering for Low Cost INS/GPS," *Proceedings of the 15th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2002)*, Portland, OR, September 2002, pp. 1143–1147.
12. Goodall, C., Niu, X., and El-Sheimy, N., "Intelligent Tuning of a Kalman Filter for INS/GPS Navigation Applications," *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007)*, Fort Worth, TX, September 2007, pp. 2121–2128.
13. Martin, H. F., Groves, P. D., Newman, M., and Faragher, R., "A New Approach to Better Low-Cost MEMS IMU Performance using Sensor Arrays," *Proceedings of the 26th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2013)*, Nashville, TN, September 2013, pp. 2125–2142.
14. Martin, H. F, Groves, P. D., and Newman, M., "The Limits of In-Run Calibration of MEMS and the Effect of New Techniques," *Proceedings of the 27th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2014)*, Tampa, FL, September 2014, pp. 162–176.
15. Titterton, D., and Weston, J., *Strapdown Inertial Navigation Technology*, IEE Radar Series, London, UK & Restin, VA: Institution of Electrical Engineers, 2004.
16. Kempe, V., *Inertial MEMS: Principles and Practice*: Cambridge University Press, 2011.
17. Walter, D., Groves, P., Mason, B., Harrison, J., Woodward, J., and Wright, P., "Novel Environmental Features for Robust Multisensor Navigation," *Proceedings of the 26th International Technical Meeting of the Satellite Division of The Institute of Navigation*, Nashville, TN, September 2013, pp. 488–504.
18. Groves, P. D., Wang, L., Walter, D., Martin, H. F., Voutsis, K., and Jiang, Z., "The Four Key Challenges of Advanced Multisensor Navigation and Positioning," *Proceedings of IEEE/ION PLANS 2014*, Monterey, CA, May 2014, pp. 773–792.
19. Brown, R. G., and Hwang, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering with MATLAB Exercises*, 4th ed.: Wiley, 2012.
20. Maybeck, P. S., *Stochastic Models, Estimation, and Control*, ser. Mathematics in Science and Engineering, New York, London: Elsevier Science, 1982.
21. Wagner, J. F., Kasties, G., and Klotz, M., "An Alternative Approach to Integrate Satellite Navigation and Inertial Sensors," *Proceedings of the 1997 National Technical Meeting of The Institute of Navigation*, Santa Monica, CA, January 1997, pp. 141–150.
22. Dissanayake, G. et al., "The Aiding of a Low-Cost Strapdown Inertial Measurement Unit using Vehicle Motion Constraints for Land Vehicle Applications," *IEEE Transactions on Robotics and Automation* Vol. 17, No. 5, 2001, pp. 731–747.

23. *"DJI Phantom Specifications"* (Available from: http://www.dji.com/product/phantom/spec) [accessed January 27, 2015].
24. Groves, P. D. et al., "Inertial Navigation versus Pedestrian Dead Reckoning: Optimizing the Integration," *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007)*, Fort Worth, TX, September 2007, pp. 2043–2055.
25. Zampella, F., Khider, M., Robertson, P., and Jimenez, A., "Unscented Kalman Filter and Magnetic Angular Rate Update (MARU) for an Improved Pedestrian Dead-Reckoning," *Proceedings of IEEE/ION PLANS 2012*, Myrtle Beach, SC, April 2012, pp. 129–139.
26. Voutsis, K., Groves, P. D., Ford, C., and Holbrow, M., "The Importance of Human Motion for Simulation Testing of GNSS," *Proceedings of the 27th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2014)*, Tampa, FL, September 2014, pp. 440–453.
27. *BMA180 Digital, Triaxial Acceleration Sensor*, Bosch Sensortec, December 2010. BST-BMA180-DS000-07.
28. *L3G4200D MEMS Motion Sensor: Ultra-Stable Three-Axis Digital Output Gyroscope Datasheet*, STMicroelectronics, December 2010, Doc ID 17116 Rev. 3.
29. *MPU-9150 Product Specification*, InvenSense Inc., 1197 Borregas Ave, Sunnyvale, CA 94089, May 2012. PS-MPU-9150A-00, Revision 4.0.
30. *MTi-G User Manual and Technical Documentation*, Xsens Technologies B.V., October 2010. Document MT0137P Rev. H.
31. *Everything About STMicroelectronics' 3-Axis Digital MEMS Gyroscopes*, STMicrotronics, July 2011. Doc ID 022032 Rev 1.
32. Tanenhaus, M., Geis, T., Carhoun, D., and Holland, A., "Accurate Real Time Inertial Navigation Device by Application and Processing of Arrays of MEMS Inertial Sensors," *Proceedings of IEEE/ION PLANS 2010*, Indian Wells, CA, May 2010, pp. 20–26.
33. Yuksel, Y., El-Sheimy, N., and Noureldin, A., "Error Modeling and Characterization of Environmental Effects for Low Cost Inertial MEMS Units," *Proceedings of IEEE/ION PLANS 2010*, Indian Wells, CA, May 2010, pp. 598–612.