

The Limits of In-Run Calibration of MEMS and the Effect of New Techniques

Henry Martin, Paul Groves
University College London, UK
Mark Newman
BAE Systems Advanced Technology Centre, UK

BIOGRAPHY

Henry Martin is a PhD student at University College London (UCL) in the Engineering Faculty's Space Geodesy and Navigation Laboratory (SGNL). His project is concerned with improving navigation performance from a low-cost MEMS IMU. He is interested in inertial navigation, IMU error modelling, multi-sensor integration and calibration algorithms. He holds a Master of Mathematics (MMath) degree from Trinity College at the University of Oxford and an MSc in Advanced Mechanical Engineering from Cranfield University. (henry.martin.11@ucl.ac.uk)

Dr Paul Groves is a Lecturer (academic faculty member) at UCL, where he leads a program of research into robust positioning and navigation and is a member of SGNL. He joined in 2009, after 12 years of navigation systems research at DERA and QinetiQ. He is interested in all aspects of navigation and positioning, including multi-sensor integrated navigation, improving GNSS performance under challenging reception conditions, and novel positioning techniques. He is an author of more than 60 technical publications, including the book *Principles of GNSS, Inertial and Multi-Sensor Integrated Navigation Systems*, now in its second edition. He is a Fellow of the Royal Institute of Navigation (RIN) and serves as an Associate Editor of both the RIN and ION journals. He holds a bachelor's degree and doctorate in physics from the University of Oxford. (p.groves@ucl.ac.uk)

Mark Newman graduated with a master's degree in electrical and electronic engineering (MEng) from the University of Aberdeen in 2001. Since then he has worked at BAE Systems and is now a Principal Design Engineer working on a variety of cutting edge technologies and future capabilities. His current field of work is focused on indoor and outdoor GPS denied positioning technologies for both ground and air based platforms. This forms part of BAE Systems' NAVSOP GPS-denied positioning system. (mark.newman3@baesystems.com)

ABSTRACT

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 or Wi-Fi positioning [1]. A full set of chip-scale MEMS

accelerometers and gyros can now be bought for less than \$10, potentially opening up a wide range of new applications. However, these sensors require calibration before they can be used for navigation[2].

Higher quality inertial sensors may be calibrated "in-run" using Kalman filter-based estimation as part of their integration with GNSS or other position-fixing techniques. However, this approach can fail when applied to sensors with larger errors which break the Kalman filter due to the linearity and small-angle approximations within its system model not being valid. Possible solutions include: replacing the Kalman filter with a non-linear estimation algorithm, a pre-calibration procedure and smart array [3]. But these all have costs in terms of user effort, equipment or processing load.

This paper makes two key contributions to knowledge. Firstly, it determines the maximum tolerable sensor errors for any in-run calibration technique using a basic Kalman filter by developing clear criteria for filter failure and performing Monte-Carlo simulations for a range of different sensor specifications. Secondly, it assesses the extent to which pre-calibration and smart array techniques enable Kalman filter-based in-run calibration to be applied to lower-quality sensors.

Armed with this knowledge of the Kalman filter's limits, 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. By 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.

1 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 \$10¹, potentially bringing inertial 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].

It is possible to calibrate an IMU which has relatively small biases and other systematic errors during normal use by using a second navigation system, such as GNSS with a very basic Kalman filter (KF) integration algorithm [1, 4, 5, 6]. This is desirable as it allows the IMU's calibration to be frequently updated without any effort or input from the user whenever the other navigation technology is available, which makes the inertial navigation system (INS) more accurate when it is used to bridge signal outages. 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, as is discussed later in this paper.

One solution is to replace the Kalman filter with a non-linear estimation algorithm, such as an unscented Kalman filter [7] or a particle filter [8]. However, these increase the processing load. Other possible solutions include a pre-calibration procedure and smart array. Pre-calibration requires the user to perform a known series of manoeuvres 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 inertial measurement unit (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 [3].

In order to decide when these alternatives are necessary, we need to 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. By 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 (Section 2). Then we establish well-defined filter failure criteria (Section 3). 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 Section 4 and the results of these simulations are presented and discussed in Section 5. The conclusions and future work are discussed in

¹Invensense MPU-9150 '9-axis' IMU (+magnetometer) available from www.digikey.co.uk from £4.40, MPU-6050 '6-axis' IMU from £3.87 (based on complete reel price, approximately double that for single sensor).

Sections 7 and 8.

2 BACKGROUND

2.1 Inertial measurement units and their errors

An inertial measurement unit (IMU) is a set of (at least) three accelerometers and three gyroscopes, which are arranged so that the sensitive axes of the sensors can measure specific force and angular rate in all three dimensions. Typically the arrangement of the sensors is as a 'triad' where the sensitive axes are mutually orthogonal (similar to the Cartesian X-, Y- and Z-axes). An inertial navigation system (INS) uses the output of an IMU to calculate a dead-reckoning navigation solution.

This solution uses the inertial navigation equations where the attitude (3D orientation) solution computed by integrating the angular rate measurements then this is used to add the specific force due to the Earth's gravity, which transforms the specific force into acceleration. Then the acceleration is integrated twice to calculate changes in the position solution. This double integration means that position and velocity errors build up very quickly. Additionally the reliance on the attitude solution to remove the large specific force due to earth's gravity means that only a few degrees of error in the attitude solution can have a large effect on the navigation system's accuracy. Thus accurate measurement of both specific force and angular rate are very important in maintaining a good navigation solution.

Inertial sensors, like any real measurement devices do not measure the specific force or angular rate perfectly. The sensor's errors can be split into two categories: stochastic and systematic.

Stochastic errors are random in nature; the most significant is usually the random noise is 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 sensors.

Systematic errors are errors which vary in a way which is fixed and/or a function of the sensors input and the environmental conditions. If systematic errors can be measured then their effect can be compensated for in the IMU's output. Systematic errors include (in approximate order of importance): bias, scale factor error, cross-axis sensitivity, gyroscope g-dependent error. Bias is where an amount of specific force or angular rate is added to the output, analogously to the 'tare' function on weighing scales. Scale factor error is when the sensitivity of the sensor is incorrect, such as measuring 98% or 101% of the true quantity. Cross-axis sensitivity is when the sensor picks up inertial forces which are not applied along its sensitive axis, this includes when the axes of the triad are not aligned correctly. Gyroscope g-dependent error is when a gyroscope mistakenly measures a specific force as an angular rate. There are other higher order errors such as sensor non-linearity, about which there is more information in [1, 9, 10].

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.

In this paper, we are considering whether ‘in-run’ calibration is possible for a *model* of sensor, rather than one specific sensor. As such, rather than talking about a sensor’s errors, which could be measured, we are considering the *distribution of errors* exhibited by a model of sensor. Which can only be measured by calibrating a large number of sensors of the model in question.

2.2 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 manoeuvres 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 manoeuvres accurately when instructed, which may not be practical for all applications. The more complex this set of manoeuvres is the more of the systematic error sources can be determined, potentially allowing higher accuracy. However, as the manoeuvres become more complex the chance that the user performs them incorrectly increases. The most simple instruction-based calibration that could be performed asks the user to ‘place the IMU still on a table’, also known as a zero-velocity update, this 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 measure gyroscope and accelerometer bias, accelerometer scale factor and gyroscope g-dependent error (details of this procedure are in [3]). If the user can be relied upon to perform more complex manoeuvres (e.g. rotating slowly about particular axes in a specific order) or other equipment is available (e.g. rate-tables or temperature controlled chambers) even more systematic errors could be determined.

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 of the IMU to determine the systematic errors of the latter. 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 than when the inertial system is being used alone. In addition it is carried out with no effort from the user. However, these advantages are set against one significant disadvantage. A reasonably good initial estimate of the value of the systematic errors is required for the in run estimation to remain stable and converge towards the correct value, see Section 3.1. Further one needs to know how good this initial estimate is so that the uncertainty is modelled correctly.

Clearly one way to get around this problem is to have a hybrid of a simple instruction-based pre-calibration followed by “in-run” calibration or to use an old estimate to

initialise the “in-run” estimation, however to design such a system it is necessary to determine how good an initial estimate is required. This raises the question of how good the pre-calibration must be in order to bring the residual errors within the tolerance limits of the ‘in-run’ calibration.

2.3 INS-GNSS integration based on a Kalman-filter-derivative

Many different navigation technologies could be used to aid/calibrate the INS, maximum robustness is achieved by combining many different sensors [11], but complex multi-sensor navigation brings many challenges [12]. For the remainder of this paper we assume that the navigation system used for aiding/calibrating the INS is a global satellite navigation system (GNSS). This is the method most commonly used in practice due to the low cost of GNSS user equipment, free infrastructure and fairly high accuracy and availability. INS-GNSS integration is a well-established technique whose advantages and disadvantages are already well known [1, 5].

The Kalman filter (KF) [4] is an estimation algorithm that is linear; if the system is not it must be linearised. It works in state space, that is, each quantity estimated is a state. For example 3 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. The mean square error (MSE) of the difference between these two quantities is minimised. The filter is recursive, in that it does not store the old measurements the state estimates aggregate this information in combination with the state covariance matrix (often known as the ‘P’-matrix), this includes the filter’s estimate of state estimation error variances (the diagonal entries of the P-matrix). The entries in this matrix store how good the filter thinks its current state estimates are.

When a real systems, such as INS and GNSS, are modelled with a Kalman filter, some basic assumptions need to be made [13]:

- the state errors (residuals) have a Gaussian (normal) distribution, which makes the MSE optimisation valid;
- the noise terms are ‘white’-noise (not correlated with time), to get around this assumption the filter is told the noise is greater than it really is (over-bounding); and,
- most importantly, that the system propagation and measurements are linear combinations of states. The linearising approximations include the ‘small angle approximation’ and that the products of state estimate errors are negligible.

If these assumptions are not met 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’.

In order for the filter to behave correctly, the best possible initial estimates of the states should be used and also the filter’s covariance matrix should be correctly ini-

tialised with the ‘initial uncertainty’ associated with each state. However, often this is not known, and even when it is known the uncertainty is typically exaggerated to aid the stability of the filter. Additionally the filter needs to know how much (if at all) states are expected to change through time, and how reliable and accurate the measurements provided are. Choosing the correct level for all these settings is known as tuning, and doing it correctly is critical for proper filter performance. Unfortunately the correct settings are specific to individual applications, and often trial-and-error is the method used.

More information about Kalman filtering can be found in standard texts such as [13, 14].

3 KALMAN FILTER FAILURE

For the work presented here, where we examine the limits of Kalman filter performance, we must have some idea of when we judge a Kalman filter to be performing unacceptably, inadequately or unstably, which we will from now on call “failing”. Additionally, having decided what constitutes failure we need some criteria for whether a particular distribution passes or fails, that is, we need a test to which we can subject a simulation giving a ‘pass’ or ‘fail’. We shall discuss these two points in Sections 3.1 and 3.2, respectively.

3.1 When has a Kalman filter failed

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

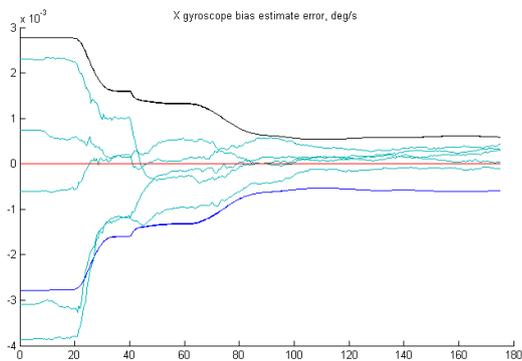


Figure 1. An example of a typical well behaved filter’s state estimate error. The filter estimates converge towards zero and their error is well described by the state uncertainty.

A ‘well-behaved’ KF should start with state uncertainties associated with its estimates of each of the states which slightly exceed the real standard deviation (SD) of the error distribution, tuned that way to aid filter stability when the true stochastic behaviour of the states diverges from the KF’s assumptions. Then as more aiding (GNSS) measurements are added both the filters state uncertainty and the real SD of the error in the state estimates should reduce together, particularly sharply when manoeuvres (such as turns) take place, the KF’s state uncertainty appearing as a slightly smoother line than the actual SD of the state estimate error. This is illustrated in Figure 1. An important

point to note is that, through time, as more data is added to the filter, the errors in the estimates get smaller, although the accuracy of the estimate will plateau at a level that depends on both the GNSS accuracy, the IMU noise and time variation of the systematic errors.

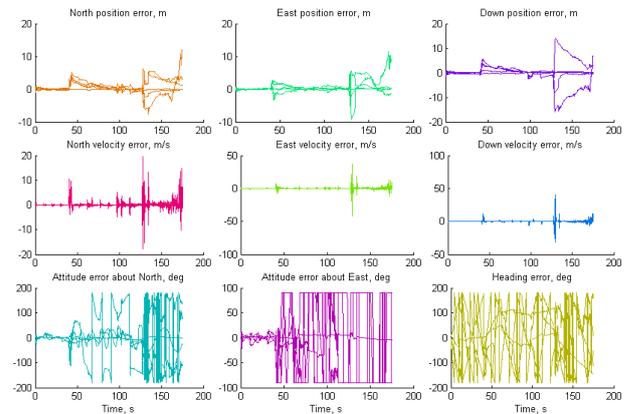


Figure 2. An example of a typical badly behaved filter’s estimate errors in its position attitude and velocity solution. Note how the filter starts fairly under control but as soon as the attitude errors about North and East exceed a few tens of degrees (around 30 seconds) then the variation of all the states is becomes extremely erratic.

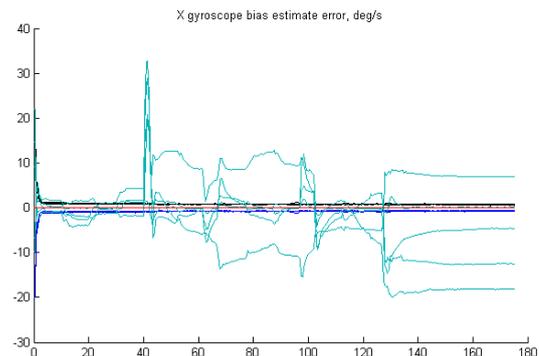


Figure 3. An example of a typical badly behaved filter’s state estimate error. The filter estimates do not converge to zero but vary erratically, additionally their variation is much greater than the state uncertainty.

When the sensor errors are very large, the linearising assumptions described in Section 2.3 break very quickly. Consider the following situation. GNSS integration can only correct attitude errors indirectly based on their effect on the position/velocity solution. So when the gyro bias is large, the attitude error, which started as a small angle, grows more quickly than it can be corrected. Rapidly the linear combinations of states that were valid when the attitude error was a ‘small angle’ cease to be valid, and for instance the corrections from the measurements are applied incorrectly. Additionally when the errors are ‘large’ then other KF assumptions break such as product of two errors being negligible. This kind of behaviour can be seen in Figure 2. This produces similarly erratic estimates for the IMU

error states, see Figure 3. Clearly, if we run a Monte Carlo simulation and the results are as erratic as in that example detecting this is a ‘failure’ is straightforward.

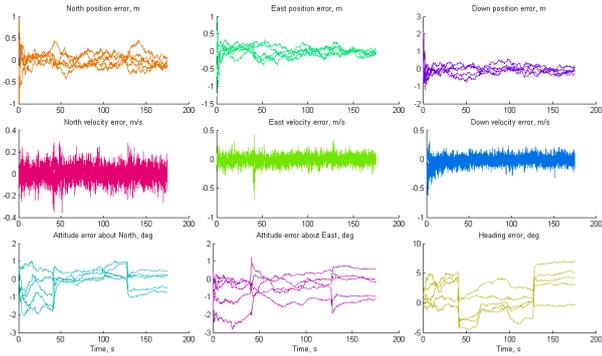


Figure 4. An example of an inconsistently behaved filter’s estimate errors in its position attitude and velocity solution. Note how the filter starts fairly under control but as soon as the attitude errors about North and East remain under 2 degrees, making the small angle approximation reasonable, although the heading error is larger.

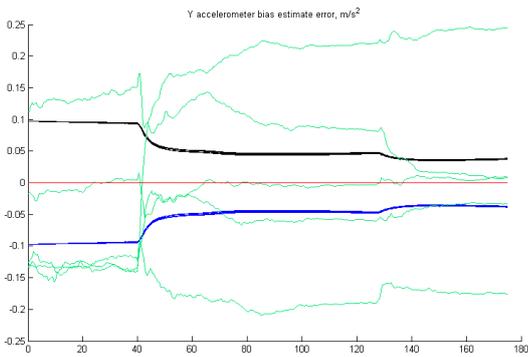


Figure 5. An example of a typical inconsistent filter’s state estimate error. Some of the filter estimates converge towards zero but others do not. The spread of estimate errors is not well described by the state uncertainty.

However, when the IMU errors are smaller the position, attitude and velocity solution can remain fairly reasonable (Figure 4) even though the IMU error estimates produced are inconsistent (Figure 5). This poor performance cannot be explained by something as simple as the small angle approximation being broken. In this situation deciding to label this Monte Carlo simulation as failed is not so straightforward.

We shall discuss the possible ways in which we could decide whether a particular simulation has failed in Section 3.2.

3.2 Kalman filter failure criteria

To decide when INS-GNSS integration has failed the context we wish to use that is important. 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 KF states corresponding to the IMUs errors (the IMU-error-states), here the accelerometer and gyroscope bias states, 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 become unstable and the error estimates go wildly wrong. Also, the reduction in the errors’ distributions would be expected to be different for sensors with different stochastic errors for example.

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 (see Section 4.4) if the simulation were longer perhaps the uncertainty in the IMU error states would continue to decrease. We could choose some arbitrary amount of time and test whether the calibration resulting from the KF was ‘adequate’ after that amount of time. However, in some application scenarios there may be tens of minutes or even hours between outages. However, what is ‘adequate’ performance in bridging a GNSS outage? Consider two INSs which have different amounts of measurement noise and both have their systematic errors perfectly calibrated. The less noisy INS would perform better, so we would need to specify the ‘adequate’ performance carefully. 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’.

In this paper we take a different approach to deciding whether a INS-GNSS integration Kalman filter is ‘successful’, we look at filter stability. Consider an INS that has high noise and extremely low bias, the KF integration may not be able to improve the estimate of the bias, but if the bias were to slowly change then, if the filter was stable, it would be able to compensate for any change. However, this situation does not occur within the parameter ranges tested. Similarly if the bias is very high it may take many minutes for the filter’s error state estimates to finish converging, but if the filter is stable then eventually there will be a good estimate of the IMU errors.

In general, a Kalman filter will be stable if all of the assumptions mentioned in Section 2.3 are valid and if the variances and covariances stored in its state 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, at least not for all the states. So this quantity cannot be tracked however, in simulations the ‘truth’ from which the estimates differ is known. This means that criteria based on the real error are possible.

A necessary but not sufficient condition for the state covariance matrix being correct is that the filter is tuned cor-

rectly, as otherwise it is either wrong from the start or will quickly become so.

The specific condition that we test is the relationship between that filter’s state uncertainty for every state at each epoch (averaged across the runs in the Monte Carlo simulation) and the root-mean-squared-error (RMSE) across the set of simulations of the filter’s state estimate error². This is then compared with two different thresholds. First that the worst state’s RMSE does not exceed 1.5σ for more than 10% of the simulation time, where σ^2 is the corresponding state variance from the state covariance matrix.

The second test is as follows. We check for every state every epoch in every run of the Monte Carlo simulation whether or not that state-estimate-error is within the corresponding 1σ uncertainty estimate from zero, then we calculate what proportion of the simulations that are outside this 1σ for each epoch and each state and if this exceeds the 32% of the simulations expected. We then count up the proportion of the time that each state spends outside these uncertainty bounds and if the worst state exceeds 25% of the time, the Monte Carlo simulation for those inputs fails.

The reason we use the worst state (rather than averaging over all states) is that we need all of the states to be stable in order for all of the IMU error states to be estimated correctly. We could have chosen thresholds that would fail the distribution as soon as they were exceeded (or exceeded 1% of the time), for instance 4 or 5σ bounds. However as the simulations are relatively short, these lower thresholds with some tolerance of being exceeded were chosen.

4 SIMULATION APPROACH

In this section the approach taken to run simulations to determine whether a particular IMU model is suitable for ‘in-run’ calibration will be outlined. First, the general approach will be outlined (Section 4.1), then the particular Kalman filter variant used will be discussed (Section 4.2). This is followed by details of the simulation algorithm’s design (Section 4.3), then the motion scenario used (Section 4.4) and the GNSS parameters (Section 4.5). Finally, the way in which the set of inputs were searched is outlined in Section 4.6.

4.1 Simulation Philosophy

The first important point about how the simulations presented in this paper are carried out is that different IMUs are modelled by how their measurements deviate from the true angular rate and specific force measurements. That is, we add errors to the output of a fictional ‘perfect IMU’ of differing magnitudes to simulate IMUs of different grades.

Furthermore, the values for each of the different systematic errors are selected randomly for each run of the Monte-Carlo simulation under the assumption that the errors are normally distributed with a zero-mean. Thus the only input needed to generate the set of systematic errors for all the runs of the Monte-Carlo simulation is the standard deviation for that systematic error. Thus the inputs for each

²This is used rather than SD to account for the fact that the real distribution may become non-zero-mean.

Monte-Carlo simulation are standard deviations of systematic errors and noise power-spectral-densities (PSDs). This assumption of zero-mean Gaussian distribution of errors is not necessarily true in reality, the implications of it will be discussed in Section 6.

In the Monte Carlo simulations for this paper we generate independent noise sequences for every one of the runs for every test, and while the GNSS measurements are selected from the same distribution each time they are selected independently for each run of each Monte-Carlo simulation. This approach maximises the randomness of the Monte Carlo simulation, which in the authors’ opinion is better where we are aiming to find general rather than specific behaviour, assuming the number of Monte-Carlo simulations is sufficiently large.

4.2 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 being to see the limits of this basic filter and then see when more complex filters were required. As such we also use the minimum realistic number of states. This means that it is both the most simple to programme 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. It is also a standard Kalman filter rather than an extended Kalman filter (EKF) [13, 15] or unscented Kalman filter (UKF) [7]. This means that it has linear system and measurement models. It also has closed-loop correction of the inertial sensor error states, which helps improve the stability of the filter if the magnitude of these states were to become large, because it is equivalent to an EKF system propagation [1].

The states modelled by this ‘basic Kalman filter’ will also be the minimum commonly used configuration. That is position, attitude and velocity (3×3), accelerometer bias (3) and gyroscope bias (3). This is a total of 15 states. If this were being extended the next most significant IMU errors that should be modelled would be either the accelerometer and gyroscope scale factor errors (2×3) or a full cross coupling and scale factor matrices (2×9) followed by the gyroscope g-dependent error matrix (9). Modelling additional states makes the filter significantly slower as the majority of the simulation’s time is spent performing matrix multiplications and inversions both of which are approximately $O(n^3)$, where the n is the number of states³.

This ‘basic’ Kalman filter is the one used for the simulations presented in this paper.

4.3 Algorithm Process

The process through which the algorithm runs for making a Monte Carlo simulation of a single distribution is shown in Figure 6. First the inputs to the code are specified, these

³Matrix inversion can in fact be reduced to $O(n^{2.807})$ [16] and in some cases to $O(n^{2.376})$ [17].

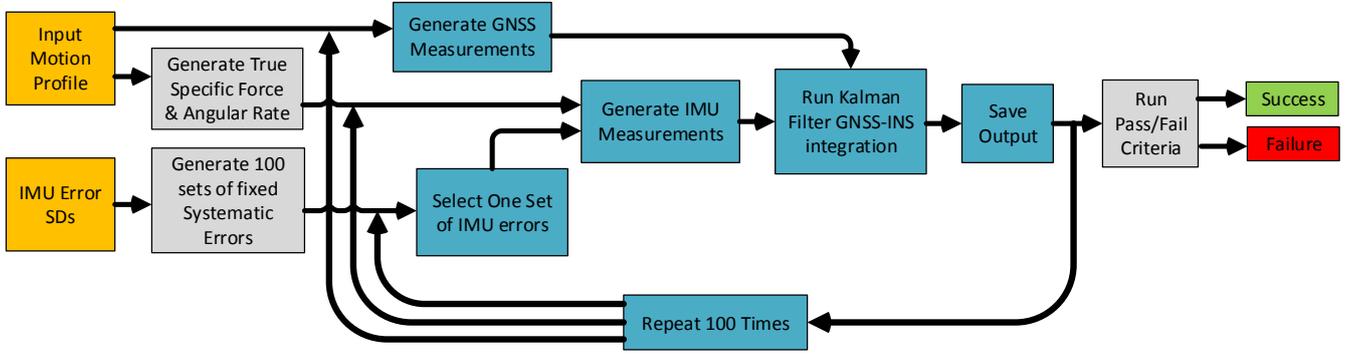


Figure 6. Flow Diagram of the Monte Carlo simulation algorithm. Inputs in yellow and Monte Carlo part in blue.

inputs and the ranges and values used are given in Table 1, however the reason that particular ranges were chosen is discussed in Section 4.6. Additionally the true motion profile selected is an input, this will be discussed in Section 4.4. Next values for the systematic IMU errors for each of

Parameter	Value or range used
size of MC simulation	100
attitude initialisation error SD	0.5 deg (all axes)
Accel. Bias mean	0 μ g
Accel. Bias SD	1000 to 100,000 μ g
Accel. Noise PSD	100 to 500 μ g/ \sqrt{Hz}
Accel. Scale factor error SD	0.06% to 3%
Accel. Cross axis sensitivity SD	0.025% to 1%
Accel. quantization level	0.01m/s ²
Gyro Bias mean	0 deg/hr
Gyro Bias SD	10 deg/hr to 20 deg/s
Gyro Noise 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

Table 1. The inputs used to the Monte Carlo Simulation, which generate the IMU errors and tune the filter. The value chosen or the range of values tested is also shown

the 100 simulation runs are chosen from a Gaussian distribution with the standard deviations (SDs) that were inputs, for example 100 accelerometer biases are chosen and 100 3×3 g-dependent error matrices. Then the Kalman filter tuning parameters are set using the input distribution SDs. As we know the true error distribution we could tune the KF precisely, but, in line with standard practice, the error distributions and noise power spectral densities (PSDs) are over-bounded in the KF integration [1]. This helps maintain filter stability, as all unmodelled errors appear as noise to the filter. Noise is input as double the actual noise PSD and initial position, velocity and attitude states uncertainties are overmodelled by a factor of 2.

The next step is to calculate the true specific force and angular rate measurements from the true motion profile and 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. Then inertial navigation equations and the basic Kalman filter INS-

GNSS integration (discussed in Section 4.2) is run for each of the 100 sets of simulated IMU and GNSS measurements. The software that does this part is available open-source on the CD accompanying [1]. 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 KFs estimate of the uncertainty that it has for each state, in the manner discussed in Section 3.2, are made. This gives the result of ‘pass’ or ‘fail’ for the particular error distribution.

4.4 Simulation Motion Scenario

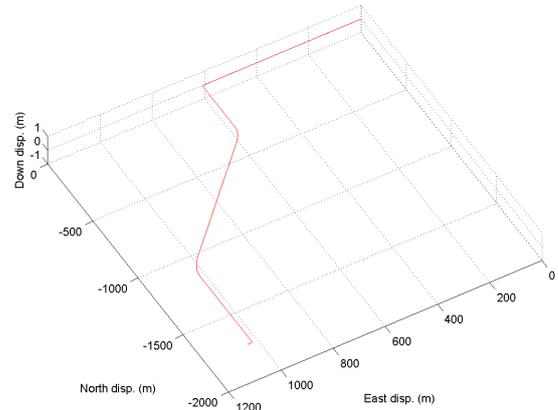


Figure 7. 3D projection of the truth motion profile used to for the Monte Carlo simulations.

In the research presented in this paper we chose to use a truth motion profile for the Monte Carlo simulations that consists of a typical car motion lasting three minutes and containing three turns. It is shown in Figure 7. This motion profile is fairly representative of the navigation scenario that a typical consumer grade IMU might be used in. However, as we do not wish to make the results specific to cars or land vehicles we do not implement land-vehicle motion constraints [18], these would reduce the INS drift and thus would make the calibration of the IMU systematic errors easier, so we are testing the more challenging scenario. Additionally the relatively limited number of different manoeuvres make it relatively difficult to separately observe

the different IMU errors in this scenario.

The reason that a pedestrian motion scenario is not used is two-fold. Firstly, realistic pedestrian motion is difficult to generate [19]. Additionally, if one was trying to use a consumer grade IMU to navigate as a pedestrian then one would much more likely use a pedestrian dead reckoning (PDR) [20, 21] approach, so calibrating the sensors to use the full inertial navigation equations does not make sense, one would more likely calibrate the PDR parameters (e.g. step-length) using the aiding navigation system.

4.5 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 signals are available.

As the simulation is relatively short (3-minutes) the variation in GNSS systematic errors has been neglected, and so the measurements used for the Kalman filter are GNSS-like generic position and velocity measurements, which are the measurement type required by the loosely-coupled integration. The associated measurement noise SD parameters are 2.5m and 0.1 m/s on each axis for the position and velocity respectively.

4.6 Restricting the Search Space

There are a large number of possible variables which could be changed to run different Monte Carlo simulations which were listed in Table 1. In order to be able to run enough simulations on limited resources the Monte-Carlo simulations contain 100 runs each. Given there are a vast number of possible combinations of the 14 possible input parameters, we made a few assumptions that could reduce the number of potential combinations. Firstly, only error quantities 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. Secondly, we have chosen to vary some but not all of the possible inputs, for instance we are fixing the sensor quantisation and the attitude initialisation error. Also, it seems very unlikely that certain combinations of errors exist, such as high cross-coupling with extremely accurate scale factor, particularly as both are partially caused by misalignment of the triad’s sensitive axes. For this reason, we vary some of the errors together, with one parameter for accelerometer ‘unmodelled’ errors encompassing, accelerometer scale factor error and accelerometer cross-coupling, and a second parameter for gyroscope unmodelled errors which combines: gyroscope scale factor error, gyroscope cross coupling and gyroscope g-dependent error. These two error parameter sets are split into three levels for testing, ‘low’, ‘medium’ and ‘high’, which are given in Table 3. We also consider all the error sources to be zero-mean, although this is not always the case in real sensors

(see Section 6).

This leaves six different distribution parameters to test. As we are trying to find the border between “success” and “failure” we are looking for a five-dimensional subspace in six dimensional space, analogously with how a surface is a two-dimensional subspace of three-dimensional space. As we had no idea of the expected structure of this subspace initially we proceeded as follows:

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) ×3 (unmodelled accel) ×3 (unmodelled gyro)). Then having identified the general structure of the space, interesting parts 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% the value of the previous point, and thus it appears to be equally spaced on a log-scale.

5 RESULTS AND DISCUSSION

Discussion of the results of the simulations shall be split into two parts. These will be when the unmodelled error parameters (scale factor, cross coupling and gyro g-dependent error) are ‘small’, that is to say insignificant; and when they are large enough to have an effect on the results.

5.1 Results when the unmodelled IMU error parameters are ‘small’

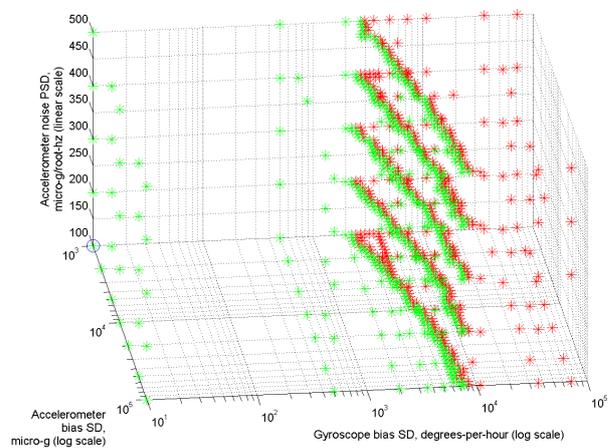


Figure 8. Three dimensional subspace of the search space showing where the KF fails for different values of accelerometer and gyroscope bias and accelerometer noise with gyroscope noise fixed at 0.01 deg per root hour and both accelerometer and gyroscope unmodelled errors fixed to ‘low’ (see Table 3).

When the unmodelled IMU error parameters are small the assumptions we made by using a Kalman filter (KF) with only 15 states is reasonable. That is, in relation to these errors the filter should be fit for purpose, and if the filter ‘breaks’ it must be for another reason. However, there are

Sensor Manufacturer Model Type	Bosch BMA180 [22] accelerometer	STMicroelectronics L3G4200D [23] gyroscope	Invensense MPU-9150 [24] single-chip IMU	Xsens MTi-G [25] factory-calibrated IMU
Accelerometer Errors				
Bias	± 60 milli-g	n/a	$\pm 80(x\&y)$ to $150(z)$ milli-g	$0.02 m/s^2$
Noise	$150 \text{ micro-g}/\sqrt{Hz}$	n/a	$400 \text{ micro-g}/\sqrt{Hz}$	0.002 to $0.004 m/s^2/\sqrt{Hz}$
Scale Factor Error	$\pm 1.5\%$ to 3%	n/a	$\pm 3\%$	$\pm 0.03\%$
Cross-Axis Sensitivity	1.75%	n/a	not specified	aligned to ± 0.1 degree
Non-Linearity	0.15 to 0.75% FS	n/a	0.5% FS	not specified
Gyroscope Errors				
Bias	n/a	± 10 to 75 dps	± 20 dps	± 1 dps
Noise	n/a	$0.03 \text{ dps}/\sqrt{Hz}$	$0.005 \text{ dps}/\sqrt{Hz}$	0.05 to $0.1 \text{ dps}/\sqrt{Hz}$
Scale Factor Error	n/a	$\pm 4\%$ [26]	$\pm 3\%$	not calibrated
Cross-Axis Sensitivity	n/a	not specified	$\pm 2\%$	aligned to ± 0.1 degree
Non-Linearity	n/a	0.2% of FS	0.2% FS	not specified

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 \$2500. Where a range is given this parameter depends on the full-scale measurement range selected.

Parameter	‘low’	‘medium’	‘high’
Accelerometer			
Scale factor error SD	0.06%	1%	3%
Cross axis sensitivity SD	0.025%	0.5%	1%
Gyroscope			
Scale factor error SD	0.03%	1%	3%
Cross axis sensitivity SD	0.02%	1%	2%
g-dependent error SD (deg/hour/g)	1	10	100

Table 3. The three levels of unmodelled IMU errors tested

error distributions for which the KF fails even with ‘low’ unmodelled errors. These are illustrated in Figures 8, 9 and 10. As it is impractical to present 6-dimensional diagrams, we present 3-dimensional diagrams which show the variations in 3 parameters when the 3 parameters are 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 under 0.6 degree per second (dps), equivalent to 2,210 degrees per hour (deg/hr), gyroscope bias standard deviation (SD) were successful and all those tested above 3 dps (10,800 deg/hr) were failures. This makes it clear that the integrity of the attitude solution is key to INS-GNSS KF stability. This is not surprising as while the GNSS updates directly correct position and velocity errors, the attitude solution corrections rely on indirect corrections from its effect on position and velocity, particularly its use to compensate for the specific force due to gravity. Additionally the small angle approximation required to linearise the KF is not valid when the attitude solution is poor. It is interesting, although not surprising, to see that this failure point is between the specified performance of factory calibrated IMUs and consumer-grade MEMS gyroscopes (see Table 2).

Variation of the ‘border’ between success and failure

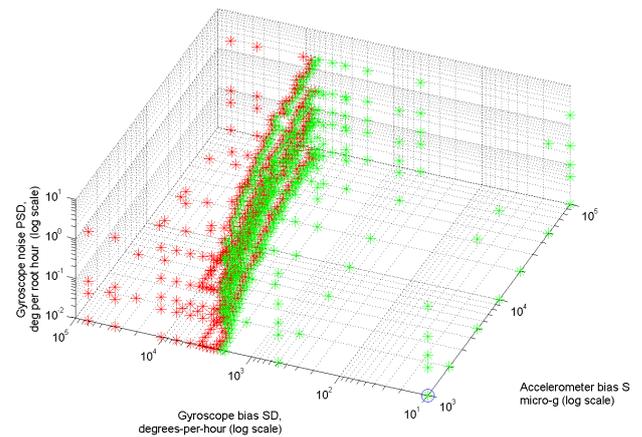


Figure 9. Three dimensional subspace of the search space showing the where the KF fails for different values of accelerometer and gyroscope bias and gyroscope noise with accelerometer noise fixed at $100 \mu g/\sqrt{Hz}$ hour and both accelerometer and gyroscope unmodelled errors fixed to ‘low’ (see Table 3).

can also be observed, which depends on the other IMU error parameters. The most significant variation is shown to be with accelerometer bias, having an effect of approximately factor 3.5 on the maximum possible gyroscope bias (2210 - 7717 deg/hr) from one end of its range to the other (1-100 milli-g), this effect is still small (see Figure 8). It is perhaps surprising that worse accelerometer bias performance allows the filter to cope with more gyroscope bias, as one might perhaps have expected the effect to be the other way around. A possible explanation is that the larger accelerometer bias uncertainty in the Kalman filter allows more scope to absorb the unmodelled errors.

The effect of the IMU’s two noise parameters on the position of the boundary is even smaller. In Figure 8 one can

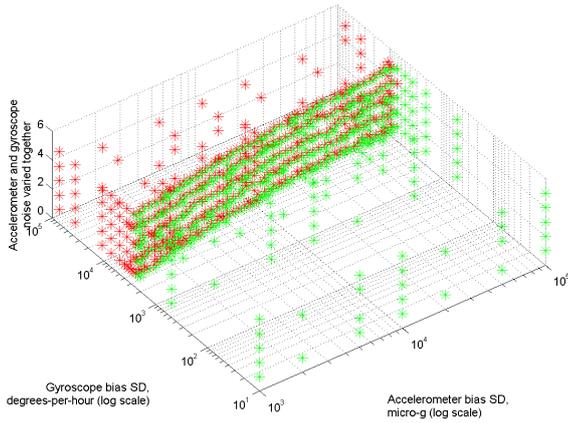


Figure 10. Three dimensional subspace of the search space showing the where the KF fails for different values of accelerometer and gyroscope bias and accelerometer and gyroscope noise varying together and both accelerometer and gyroscope unmodelled errors fixed to ‘low’ (see Table 3).

observe the slight change in the boundary as accelerometer noise is increased (z-axis of Figure 8) of only a factor of 1.2-1.5 (at most) and the effect of the gyroscope noise (Figure 9) is less than the grid spacing (geometrically spaced at 110%), even varying both noise parameters together (Figure 10) does not show a more significant effect.

This leads to the conclusion that the gyroscope bias SD really dominates the KF stability when the unmodelled errors are ‘low’.

5.2 The effect of the un-modelled IMU error parameters on the results

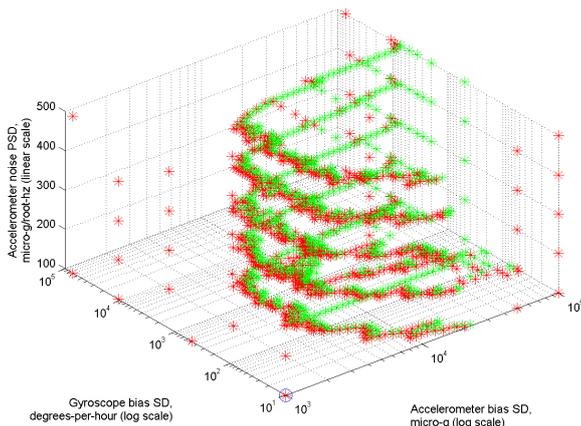


Figure 11. Three dimensional subspace of the search space showing the where the KF fails for different values of accelerometer and gyroscope bias and accelerometer noise with gyroscope noise fixed at 0.01 deg per root hour and both accelerometer and gyroscope unmodelled errors fixed to ‘medium’ and ‘low’, respectively (see Table 3).

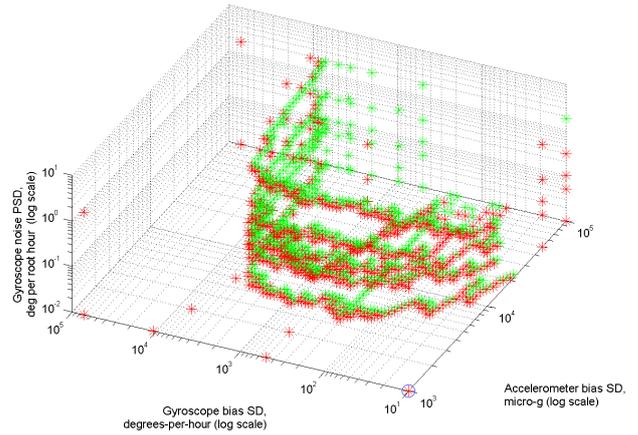


Figure 12. Three dimensional subspace of the search space showing the where the KF fails for different values of accelerometer and gyroscope bias and gyroscope noise with accelerometer noise fixed at $100 \mu g/\sqrt{Hz}$ hour and both accelerometer and gyroscope unmodelled errors fixed to ‘medium’ and ‘low’, respectively (see Table 3).

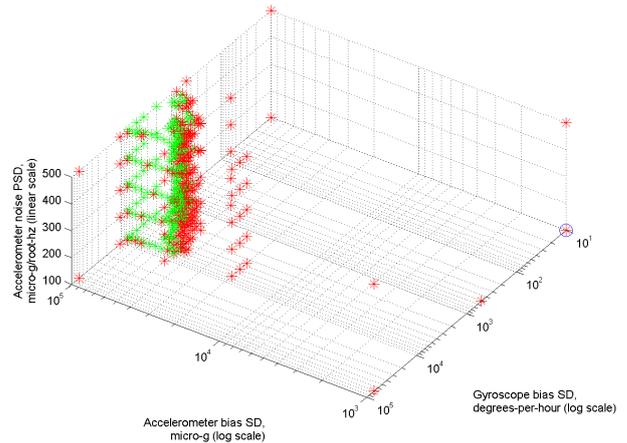


Figure 13. Three dimensional subspace of the search space showing the where the KF fails for different values of accelerometer and gyroscope bias and accelerometer noise with gyroscope noise fixed at 0.01 deg per root hour and both accelerometer and gyroscope unmodelled errors fixed to ‘low’ and ‘medium’, respectively (see Table 3).

When we vary the parameters which are not modelled 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 mentioned in Section 4.6, we only tested 3 levels of un-modelled errors of the gyroscopes and accelerometers, and we varied them together so ‘low’ scale factor errors were not combined with ‘high’ cross axis sensitivity. However, unmodelled errors of the gyroscopes and the accelerometers were tested separately. These three levels are shown in Table 3. In Section 5.1, the results for when

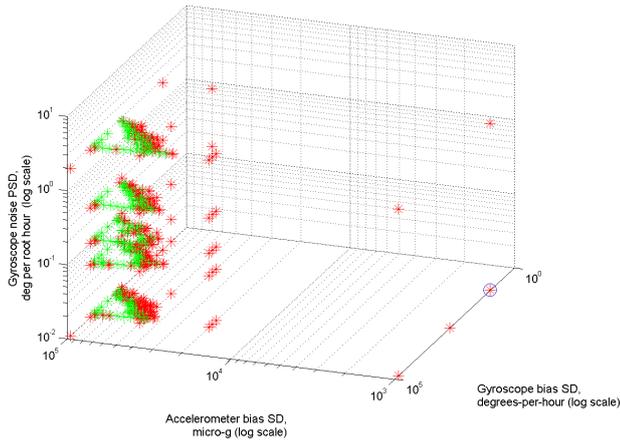


Figure 14. Three dimensional subspace of the search space showing the where the KF fails for different values of accelerometer and gyroscope bias and gyroscope noise with accelerometer noise fixed at $100 \mu\text{g}/\sqrt{\text{Hz}}$ hour and both accelerometer and gyroscope unmodelled errors fixed to ‘low’ and ‘medium’, respectively (see Table 3).

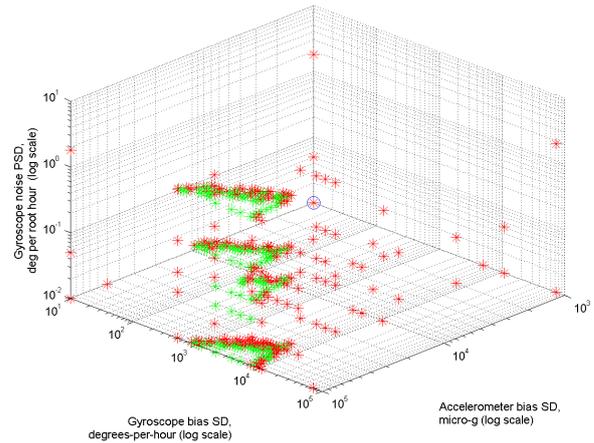


Figure 16. Three dimensional subspace of the search space showing the where the KF fails for different values of accelerometer and gyroscope bias and gyroscope noise with accelerometer noise fixed at $100 \mu\text{g}/\sqrt{\text{Hz}}$ hour and both accelerometer and gyroscope unmodelled errors fixed to ‘medium’ (see Table 3).

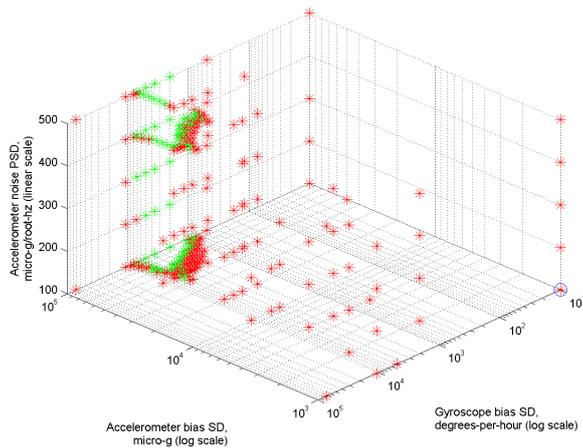


Figure 15. Three dimensional subspace of the search space showing the where the KF fails for different values of accelerometer and gyroscope bias and accelerometer noise with gyroscope noise fixed at $0.01 \text{ deg per root hour}$ and both accelerometer and gyroscope unmodelled errors fixed to ‘medium’ (see Table 3).

both of these parameters were ‘low’ were presented. Here we present the results when one or both of the gyroscopes and accelerometers unmodelled error parameters were set to ‘medium’ or ‘high’.

Firstly, of the error distributions tested that had ‘high’ unmodelled gyroscope errors none ‘passed’, and there were only a few successful distributions with ‘high’ unmodelled accelerometer errors, and these all occur when the accelerometer bias is $100,000 \mu\text{-g}$ and the gyroscope bias is 1 or 2 dps, that is when both are very high.

When the unmodelled errors are at the ‘medium’ level, the results are more instructive. The results for when the

accelerometer unmodelled errors are ‘medium’ and the gyroscope unmodelled errors ‘low’ are shown in Figures 11 and 12, with different quantities varying on the z-axis: accelerometer noise and gyroscope noise respectively. Unsurprisingly the ‘success-failure’ border when both unmodelled errors are ‘low’ (approximately gyroscope bias SD $< 2 \text{ dps}$) is repeated from Figure 8 and 9 but another boundary is added on the other side where the gyroscope biases must be *greater than* a certain amount and this amount decreases with increasing gyroscope or accelerometer noise. This is particularly visible in Figure 12.

When the accelerometer unmodelled errors are ‘low’ and the gyroscope unmodelled errors ‘medium’ the results are shown in Figures 13 and 14, with different quantities varying on the z-axis: accelerometer noise and gyroscope noise respectively. Similarly here there ‘success area’ is at the high-bias corner of the equivalent area when both unmodelled errors were ‘low’. The size of this area is noticeably smaller than the equivalent when the accelerometer errors were ‘medium’, and similarly the size of the area grows with increasing gyroscope or accelerometer noise. This is particularly visible in Figure 13.

When both the accelerometer and gyroscope unmodelled errors are ‘medium’ the results are shown in Figures 15 and 16, with different quantities varying on the z-axis: accelerometer noise and gyroscope noise respectively. In this case the area is similar to the previous two cases but even smaller, and it only grows slightly with more gyroscope or accelerometer noise.

We aim to explain this behaviour by considering two things.

First, consider how the unmodelled errors appear to the filter. While the angular rates and horizontal specific forces (accelerations) roughly average to zero across the whole simulation, the vertical (z) specific force averages in-

stead to roughly $9.8ms^{-2}$ due the earth gravity. Consider the scale-factor-and-cross-coupling matrix, 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

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

The unmodelled errors in the third column are those depending on the z-axis specific force. As a result these errors appear (on average) to the simplified IMU model used in the basic Kalman filter (see Section 4.2) as an additional bias and noise on the signal, whereas the other six errors only appear as noise. In the equivalent M-matrix for the gyroscope all 9 errors must be absorbed by the measurement noise. However, for the gyroscope g-dependent error matrix, which is another 3×3 matrix relating the effect of the specific force in x, y and z on the angular rate readings in x, y and z, 3 of the nine entries appear as an additional bias.

The second thing to consider is that during the KF tuning additional measurement noise was not added to account for this phenomenon. This was a deliberate choice as we aimed to see when these states needed to be modelled.

In order to absorb the unmodelled IMU error parameters there needs to be both ‘enough’ overmodelling of the bias states to cover the ‘extra bias’ from the unmodelled errors and ‘enough’ overmodelling of the noise was needed to cover the ‘extra noise’. Given that the automatic filter tuning used here uses a constant multiple of the actual noise/bias to configure the KF for the measurement noise/bias state uncertainty (respectively), when the noise/bias is higher the overmodelling is also higher in absolute terms, as it is fixed in proportion of the noise/bias. This would explain why the filter can cope with higher unmodelled errors when the biases/noise are higher.

It would be possible without having any significant effect on the computational or requirements or the difficulty of writing the filter, to add both additional bias uncertainty and increase the system noise to try to account for the unmodelled IMU error parameters, which is effectively modelling their ‘average effect’. If this were done correctly then this might have a positive effect on filter stability. This would be an interesting extension of this research. However, it should be noted that overmodelling error parameters has drawbacks. Over-modelling the state uncertainties might increase the convergence time as the state would ‘start’ further from the truth estimate. It also might worsen the observability of some error states. Over-modelling the system noise parameter may also slow the convergence time as well as reducing the accuracy of the estimate after convergence.

6 PRACTICAL APPLICATIONS OF THESE RESULTS

6.1 Basic application

One of the aims of this research is to produce a resource where navigation system designers can look up the specifications of the IMU they are planning to use which determines 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.

As an example of how this could be applied, the real sensor models, whose specifications were given in Table 2, are plotted in Figure 17 as different coloured 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’’ [25], is very close to the ‘low’ unmodelled errors tested here. As such these results suggest that basic KF integration with the Xsens should be stable, or at least it would be if the tuning was handled carefully.

6.2 Arrays and Smart Arrays

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 the noise and the distribution of the fixed errors. Consider a sensor model where a particular systematic error is distributed normally with mean 0 and standard deviation σ . If one sensor is used in the IMU then this systematic error will be distributed with standard deviation σ . However, if the output of n sensors are combined in the IMU then that systematic error will be distributed for the IMU with standard deviation $\frac{\sigma}{\sqrt{n}}$ and mean zero⁴. This applies to all the systematic errors and also the mean of the output noise will be divided by \sqrt{n} and even the effect of quantisation noise⁵

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. A smart array is an inertial measurement unit (IMU) comprising an array of inertial sensors 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 [3]. When the assumptions mentioned are not met, a smart array will perform considerably better than a standard array. For example, if the bias does not have a zero-mean distribution then an anti-parallel arrangement of the sensor triads that make up the IMU will make the combined smart array’s bias zero mean [3] as well as removing most of the temperature dependent bias variation [27]. If the sensor specification is symmetric about zero (e.g. bias is ± 50 milli-g) and it is not actually zero-mean it must have a lower actual SD to fit within specifications than if it were zero-mean, so the smart array would increase ac-

⁴This is because the expectation of a sum of zero mean random variables is zero, and also the variance of the sum of independent random variables is sum of their variances so the variance of their sum is $n\sigma^2$. The mean is the sum multiplied by $\frac{1}{n}$, so the variance of the mean is the variance of the sum multiplied by $\left(\frac{1}{n}\right)^2$ that is $\frac{n\sigma^2}{n^2} = \frac{\sigma^2}{n}$ so the standard deviation of their mean is $\frac{\sigma}{\sqrt{n}}$.

⁵Assuming the raw sensor bias SD \gg quantisation level, but this will always be the case in practice.

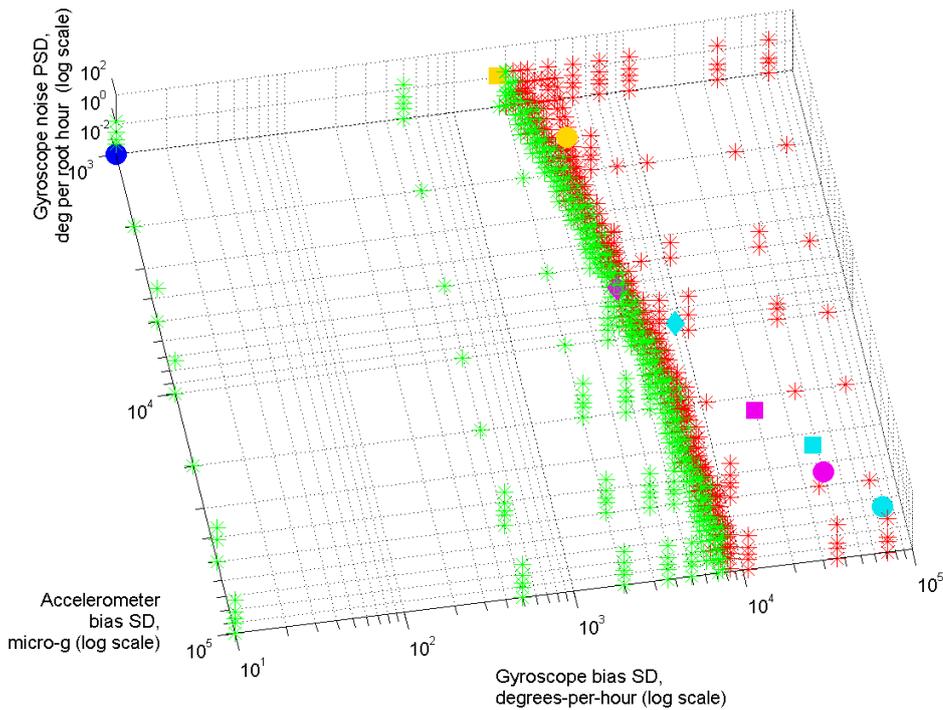


Figure 17. Three dimensional subspace of the search space showing the where the KF fails for different values of accelerometer and gyroscope bias and gyroscope noise with accelerometer noise fixed at $100 \mu\text{g}/\sqrt{\text{Hz}}$ and both accelerometer and gyroscope unmodelled errors fixed to ‘low’ (see Table 3). Also shown are the specifications of selected MEMS sensors: a tactical grade IMU, Xsens Mti-G, Invensense MPU-9150, and an IMU comprising a STMicrotronics L3G4200D gyroscope and a Bosch BMA-180 accelerometer. 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).

curacy.

Smart array techniques are important because real sensors’ errors are not distributed with perfect zero-mean normal distributions, for examples see [26] and [3]. By modelling errors as zero-mean normal distributions, we are forced to over-model the state uncertainty, which has negative implications as discussed at the end of Section 5.2.

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 flaws. It is also likely that sensors from the same production batch will have more correlated errors than others of the same design.

The effects of using a sensor array can be seen in Figure 17. The circles represent single sensors but a array of 4 of these would have half as great systematic errors and half as great a noise PSD as illustrated by the squares, and array of 16 would have a quarter the level of noise and systematic errors. There is a limit on the number of sensors that can reasonably be combined into an array, so as an example of a large, but still practical (for instance [28] has an array of 100 sensors), array the performance of 64 sensors are shown as diamonds. It can be seen that the effect of even a small array moves the Xsens sensor well inside the boundary, but even with the large array array of combined Bosch accelerometers and ST gyroscopes has only

just made it to the boundary, and the Invensense MPU-9050 array is still far away. This implies that even when using a large array some pre-calibration may be required of the raw sensors. However, only gyroscope bias need be calibrated (e.g. by static calibration) as the other errors are within the area where GNSS-calibration is possible.

6.3 Other Implications

When considering the widest implications of the research presented in this paper first we have to consider how it is limited.

First, can one generalise results from a ‘car’ motion profile to other uses of MEMS IMUs. Consider these two points: higher average velocities during the simulation help the GNSS-INS integration because GNSS position noise (close to independent of speed) has less of an effect on the overall profile when the distance travelled between GNSS epochs is greater; and, second, higher peak accelerations and angular speeds together with more variation in the true specific force and angular rate profile will excite the unmodelled error sources (Scale factor error, cross coupling error) to a greater extent. The ‘car’ motion profile used for this paper clearly has relatively high average speeds and relatively low ‘dynamics’ in terms of angular rate particularly.

Considering these two points it should be clear that applying these results to pedestrian navigation is not appropriate, however as was discussed in Section 4.4, pedestrian navigation is commonly carried out with step-detection

rather than the inertial navigation equations. The other major application of MEMS inertial navigation is low-cost autonomous vehicles and unmanned aerial vehicles (UAVs), land-based autonomous vehicles have motion characteristics very similar to cars and low-cost UAVs also travel at similar speeds but do exhibit more variation in their motion profile. It seems reasonable to assume that for these uses the results could be considered similar enough.

Secondly we have considered the simplest possible INS-GNSS integration Kalman filter possible, as it is currently configured it is not appropriate for ‘medium’ unmodelled error terms, and this ‘medium’ is pretty small in MEMS terms, although it is possible that relaxing the filter tuning may regain stability at the cost of convergence speed and accuracy. However, with array techniques as mentioned in Section 6.2, the unmodelled error parameters could be reduced by at least one ‘level’, mitigating this effect. We also might assume that if these unmodelled states were modelled then the performance is quite similar to when they were ‘insignificant’.

Bearing these limits in mind we can make a couple of points that are important for in general. The first is that basic KF integration can be sufficient to calibrate all the other INS errors if the gyroscope bias SD is below the level of around 1-2 dps. 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. 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.

Secondly, a major issues for calibration of MEMS sensors is that the calibration parameters vary with temperature. For example, the L3G4200D claims performance bias variation with temperature of 0.03 dps/°C. However, even with a 30°C operating range this is less than 1 dps. 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 would allow the INS-GNSS integration to compute the new bias.

7 CONCLUSION

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.

This approach uses criteria which are best suited to determining Kalman filter failure in this context. These criteria monitor filter stability rather than using simulated outages or thresholds for the improvement of error states.

An approach to simulating INS calibration using a basic INS-GNSS Kalman filter integration has been presented, which allows different IMU specification levels to run through integration Kalman filters using a Monte Carlo simulation method and then using the criteria mentioned above a decision can be made about whether that sensor specification is good enough for ‘in-run’ calibration. It is important to note that this technique is not specific to GNSS integration, another aiding technology could be used instead or even Zero-velocity updates.

Experiments run using this technique have been able

to show 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 0.6 and 2.5 dps depending on the accelerometer noise level, gyroscope noise level and accelerometer bias SD, the latter being the most important.

Using this most basic Kalman filter integration, it is clear that the errors which are not modelled (e.g. scale factor and cross-coupling errors), are present at a level in MEMS IMUs that is too significant to be neglected and they must be taken into account at least during filter tuning.

This paper has presented at what point in terms of the quality of the IMU a basic Kalman filter can be used. This means that navigation system designers can determine for which IMUs specification levels they need to use a more complex non-linear Kalman filter variant.

As well as providing a design resource, the outputs of this study can support the prioritization of research into ultra-low-cost inertial navigation between non-linear 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.

8 FUTURE WORK

There are a few extensions to this work that would have been made if more time and computational resources were available (in increasing order of resources required). One would have been to test more intermediate levels of the ‘unmodelled error’ parameters than ‘high’, ‘medium’, and ‘low’ accelerometer and ‘high’, ‘medium’, and ‘low’ gyroscope unmodelled errors. This would have enabled us to determine more exactly when these needed to be accounted for.

Similarly it would be interesting to verify that filter stability can be maintained with ‘medium’ and/or ‘high’ unmodelled errors by altering the filter tuning. Finally it would be good to test different levels of ‘overmodelling’ for all states and its effect on stability, specifically to see how far the .

In further research on a similar vein the stability limits of unscented Kalman filter based INS-GNSS calibration should be examined.

ACKNOWLEDGEMENTS

Henry Martin is jointly funded by the Engineering and Physical Sciences Research Council (EPSRC) and BAE Systems Advanced Technology Centre.

The authors would like to thank Santosh Bhattari (UCL) for his help explaining some of the more complex aspects of Kalman filtering and Tuula Eriksson (UCL) for her help with proofreading. Additional thanks to Martin Vermeer (Aalto University) for making this L^AT_EX template available online.

REFERENCES

- [1] P. D. Groves, *Principles of GNSS, inertial, and multi-sensor integrated navigation systems*, 2nd ed. Artech

- House, 2013.
- [2] P. Aggarwal, Z. Syed, A. Noureldin, and N. El-Sheimy, *MEMS-based Integrated Navigation*. Artech House, 2010.
- [3] H. F. S. Martin, P. D. Groves, M. Newman, and R. Faragher, "A new approach to better low-cost MEMS IMU performance using sensor arrays," in *Proc. ION GNSS*, September 2013.
- [4] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME—Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
- [5] D. B. J. Cox, "Integration of GPS with inertial navigation systems," *Navigation: JION*, vol. 25, no. 2, pp. 236–245, 1978.
- [6] J. Farrell, *Gnss Aided Navigation & Tracking: Inertially Augmented Or Autonomous*. American Literary Press, 2007.
- [7] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proc. AeroSense: The 11th Int. Symp. on Aerospace Defence Sensing, Simulation and Controls, SPIE*, 1997.
- [8] F. Gustafsson *et al.*, "Particle filters for positioning, navigation and tracking," in *IEEE Trans. on Signal Processing*, 2002, pp. 425–437.
- [9] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology*, ser. IEE Radar Series. Institution of Electrical Engineers, 2004.
- [10] V. Kempe, *Inertial MEMS: Principles and Practice*. Cambridge University Press, 2011.
- [11] D. Walter, P. Groves, B. Mason, J. Harrison, J. Woodward, and P. Wright, "Novel environmental features for robust multisensor navigation," in *Proceedings of the 26th International Technical Meeting of The Satellite Division of the Institute of Navigation*, Nashville, TN, September 2013, pp. 488–504.
- [12] P. D. Groves, L. Wang, D. Walter, H. F. S. Martin, K. Voutsis, and Z. Jiang, "The four key challenges of advanced multisensor navigation and positioning," in *IEEE/ION PLANS*, Monterey, CA, May 2014.
- [13] R. G. Brown and P. Y. C. Hwang, *Introduction to random signals and applied Kalman filtering*. New York: J. Wiley, 1992, rev. ed. of: *Introduction to random signal analysis and Kalman filtering*. 1983.
- [14] P. S. Maybeck, *Stochastic Models, Estimation, and Control*, ser. Mathematics in Science and Engineering. Elsevier Science, 1982.
- [15] J. F. Wagner, G. Kasties, and M. Klotz, "An alternative approach to integrate satellite navigation and inertial sensors," in *Proc. ION NTM*, Santa Monica, CA, January 1997, pp. 141–150.
- [16] V. Strassen, "Gaussian elimination is not optimal," *Numer. Math.*, vol. 13, pp. 354–356, 1969.
- [17] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," *J. Symbolic Comput.*, vol. 9, pp. 251–280, 1990.
- [18] G. Dissanayake *et al.*, "The aiding of a low-cost strapdown inertial measurement unit using vehicle motion constraints for land vehicle applications," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 5, pp. 731–747, 2001.
- [19] K. Voutsis, P. D. Groves, C. Ford, and M. Holbrow, "The importance of human motion for simulation testing of GNSS," in *Proc. ION GNSS*, September 2014.
- [20] P. D. Groves *et al.*, "Inertial navigation versus pedestrian dead reckoning: Optimizing the integration," in *Proc. ION GNSS*, 2007.
- [21] F. Zampella, M. Khider, P. Robertson, and A. Jimenez, "Unscented Kalman filter and magnetic angular rate update (MARU) for an improved pedestrian dead-reckoning," in *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, April, pp. 129–139.
- [22] *BMA180 Digital, triaxial acceleration sensor*, Bosch Sensortec, December 2010, bST-BMA180-DS000-07.
- [23] *L3G4200D MEMS motion sensor: ultra-stable three-axis digital output gyroscope datasheet*, STMicroelectronics, December 2010, Doc ID 17116 Rev. 3.
- [24] *MPU-9150 Product Specification*, InvenSense Inc., 1197 Borregas Ave, Sunnyvale, CA 94089, May 2012, pS-MPU-9150A-00, Revision 4.0.
- [25] *MTi-G User Manual and Technical Documentation*, Xsens Technologies B.V., October 2010, document MT0137P Rev. H.
- [26] *Everything about STMicroelectronics' 3-axis digital MEMS gyroscopes*, STMicroelectronics, July 2011, doc ID 022032 Rev 1.
- [27] Y. Yuksel, N. El-Sheimy, and A. Noureldin, "Error modeling and characterization of environmental effects for low cost inertial MEMS units," in *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, may 2010, pp. 598–612.
- [28] M. Tanenhaus, T. Geis, D. Carhoun, and A. Holland, "Accurate real time inertial navigation device by application and processing of arrays of MEMS inertial sensors," in *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, may 2010, pp. 20–26.