



National Technical University of Athens
School of Electrical and Computer Engineering
Division of Communication, Electronic and Information Engineering

Inertial and Magnetic Sensors: Calibration and Data Fusion with Application to Navigation

PhD Thesis

Konstantinos Papafotis

Supervisor

Pavlos Petros Sotiriadis, Professor, NTUA

Circuits & Systems Group

Athens, June 2022



**Operational Programme
Human Resources Development,
Education and Lifelong Learning**
Co-financed by Greece and the European Union





Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής & Συστημάτων Πληροφορικής

Αδρανειακοί και Μαγνητικοί Αισθητήρες:
Βαθμονόμηση και Συνδυασμός Δεδομένων με
Εφαρμογή στην Πλοήγηση

Διδακτορική Διατριβή
Κωνσταντίνος Παπαφώτης

Επιβλέπων
Πάυλος Πέτρος Σωτηριάδης, Καθηγητής, ΕΜΠ

Circuits & Systems Group

Αθήνα, Ιούνιος 2022



National Technical University of Athens
School of Electrical and Computer Engineering
Division of Communication, Electronic and Information Engineering

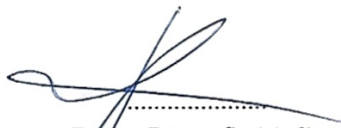
Inertial and Magnetic Sensors: Calibration and Data Fusion with Application to Navigation

PhD Thesis
Konstantinos Papafotis


Advisory Committee:

Pavlos Petros Sotiriadis Athanasios Panagopoulos Kiamal Pekmestzi
Professor, NTUA Professor, NTUA Professor Emeritus, NTUA

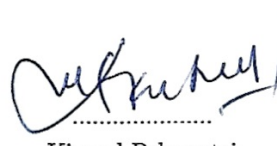
Approved by the seven-member examination committee on 15/06/2021.



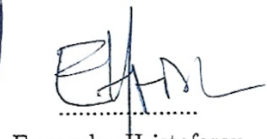
.....
Pavlos Petros Sotiriadis
Professor, NTUA



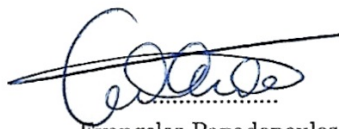
.....
Athanasios Panagopoulos
Professor, NTUA




.....
Kiamal Pekmestzi
Professor Emer. NTUA




.....
Evangelos Hristoforou
Professor, NTUA



.....
Evangelos Papadopoulos
Professor, NTUA



.....
Panayiotis Psarrakos
Professor, NTUA



.....
Ioannis Papananos
Professor, NTUA

Circuits & Systems Group
Athens, June 2022

.....

Κωνσταντίνος Παπαφώτης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Κωνσταντίνος Παπαφώτης, 2022

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς το συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Abstract

Inertial sensors (accelerometers and gyroscopes) and magnetic field sensors are widely used in a broad variety of applications. High accuracy sensors are nowadays available in chip-form including all the necessary electronic circuits for the digitization and in many cases the digital processing of the measurements. This fact enables the wider use of inertial and magnetic field sensors in several applications, ranging from low-cost commercial products (smartphones, activity trackers, alarm systems etc.) to high-end industrial and military devices.

This thesis deals with the calibration and the alignment between the sensitivity axes of inertial and magnetic field sensors. In addition, it presents two applications where inertial and magnetic field sensors are used for navigation and tracking purposes.

In both the cases of inertial and magnetic field sensors, a calibration procedure is mandatory for every sensor unit when accuracy is required. This work focuses on low-cost, three-axis inertial and magnetic field sensors and explores algorithms and methodologies for calibration and axes alignment.

In the first part, algorithms for the calibration and the alignment of the sensitivity axes of inertial and magnetic field sensors are presented. The proposed algorithms are based on optimization techniques and are designed to be computationally efficient while their application requires no special piece of equipment or external references.

In the second part, the importance of sensors' calibration is highlighted, and two applications of inertial and magnetic field sensors are presented. First, an inertial navigation system dedicated to pedestrian navigation is presented. It combines data from inertial and magnetic field sensors mounted on the shoe of a walking person and derives a long-term accurate estimation of his speed, orientation, and position.

Finally, an inertial measurement unit architecture including several three-axis accelerometers and a single three-axis gyroscope mounted on a rigid body is presented. The proposed, closed-loop system dynamically compensates for the accelerometers' measurements error and provides a high-accuracy, low-noise estimation of the specific force and the angular velocity.

Keywords: accelerometer, magnetometer, gyroscope, inertial sensors, magnetic field sensors, calibration, axes alignment, navigation, inertial measurement unit

Περίληψη

Οι αδρανειακοί αισθητήρες (αισθητήρες επιτάχυνσης και γωνιακής ταχύτητας) και οι μαγνητικοί αισθητήρες χρησιμοποιούνται ευρέως σε μεγάλο πλήθος εφαρμογών οι οποίες εκτείνονται από εμπορικά προϊόντα χαμηλού κόστους (κινητά τηλέφωνα, συστήματα συναγερού κ.λπ.) μέχρι εφαρμογές υψηλής τεχνολογίας σε τομείς όπως η βιομηχανία, η ναυτιλία κ.α. Αισθητήρες υψηλής ακρίβειας είναι πλέον διαθέσιμοι σε μορφήτσιπ μαζί με τις απαραίτητες ηλεκτρονικές διατάξεις για τη ψηφιοποίηση και πολλές φορές και τη ψηφιακή επεξεργασία των μετρήσεων κάνοντας εύκολη την ενσωμάτωση των μαγνητικών αισθητήρων σε μεγαλύτερα συστήματα.

Η παρούσα διατριβή πραγματεύεται τη βαθμονόμηση (calibration) αδρανειακών και μαγνητικών αισθητήρων καθώς και την αξιοποίηση των μετρήσεών τους σε εφαρμογές εντοπισμού θέσης (tracking) και πλοήγησης (navigation).

Τόσο στην περίπτωση των αδρανειακών αισθητήρων όσο και σε αυτή των μαγνητικών αισθητήρων, κάθε μονάδα αισθητήρα χρειάζεται μια διαδικασία βαθμονόμησης μετά την παραγωγή της προκειμένου να παράγει μετρήσεις υψηλής ακρίβειας. Η παρούσα εργασία επικεντρώνεται σε αδρανειακούς και μαγνητικούς αισθητήρες τριών αξόνων και χαμηλού κόστους και εξερευνά μεθόδους και αλγόριθμους για την καλύτερη εκμετάλλευση των δεδομένων τους σε εφαρμογές που απαιτούν υψηλή ακρίβεια, όπως για παράδειγμα η πλοήγηση.

Στο πρώτο της μέρος παρουσιάζονται μέθοδοι και αλγόριθμοι για την βαθμονόμηση αδρανειακών και μαγνητικών αισθητήρων καθώς και για την ευθυγράμμιση των αξόνων ευαισθησίας τους χωρίς την χρήση ειδικού εργαστηριακού εξοπλισμού. Οι προτεινόμενοι αλγόριθμοι στηρίζονται σε μεθόδους βελτιστοποίησης ενώ η εκτέλεση τους δεν απαιτεί σημαντικούς υπολογιστικούς πόρους.

Στο δεύτερο μέρος, αναδεικνύεται η σημασία της βαθμονόμησης σε εφαρμογές πλοήγησης και στη συνέχεια παρουσιάζονται δύο χαρακτηριστικές εφαρμογές. Συγκεκριμένα, αρχικά, παρουσιάζεται ένα σύστημα αδρανειακής πλοήγησης το οποίο συνδυάζει τα δεδομένα αδρανειακών και μαγνητικών αισθητήρων τοποθετημένων στο παπούτσι ενός πεζού ανθρώπου και παρέχει μακροπρόθεσμα μια μεγάλη ακρίβεια εκτίμησης της θέσης.

Στη συνέχεια, παρουσιάζεται ένα αδρανειακό σύστημα μέτρησης αποτελούμενο από πολλαπλούς αισθητήρες επιτάχυνσης σε συνδυασμό ένα αισθητήρα γωνιακής ταχύτητας. Το προτεινόμενο σύστημα κλειστού βρόχου αντισταθμίζει δυναμικά το σφάλμα μέτρησης των αισθητήρων επιτάχυνσης και προσφέρει μια εκτίμηση της επιτάχυνσης και της γωνιακής ταχύτητας με πολύ χαμηλά επίπεδα θορύβου.

Λεξεις Κλειδια: αισθητήρες επιτάχυνσης, μαγνητικοί αισθητήρες, αισθητήρες γωνιακής ταχύτητας, αδρανειοί αισθητήρες, μαγνητόμετρο, γυροσκόπιο, βαθμονόμηση, ευθυγράμμιση αξόνων ευαισθηρίας, πλοήγηση, εντοπισμός θέσης

Εκτεταμένη Περίληψη

Η παρούσα διατριβή πραγματεύεται τη βαθμονόμηση (calibration) αδρανειακών και μαγνητικών αισθητήρων καθώς και την αξιοποίηση των μετρήσεών τους σε εφαρμογές εντοπισμού θέσης (tracking) και πλοήγησης (navigation).

Οι αδρανειακοί αισθητήρες (αισθητήρες επιτάχυνσης και γωνιακής ταχύτητας) χρησιμοποιούνται ευρέως τα τελευταία χρόνια σε μεγάλο πλήθος εφαρμογών οι οποίες εκτείνονται από εμπορικά προϊόντα χαμηλού κόστους (κινητά τηλέφωνα, συστήματα συναγερμού κ.λπ.) μέχρι εφαρμογές υψηλής τεχνολογίας σε τομείς όπως η βιομηχανία, η ναυτιλία κ.α. Η ανάπτυξη αυτή οφείλεται σε μεγάλο βαθμό στη ραγδαία ανάπτυξη των τεχνολογιών κατασκευής αδρανειακών αισθητήρων τις τελευταίες δεκαετίες και κυρίως στην ανάπτυξη της τεχνολογίας MEM (micro-electro-mechanical) αισθητήρων η οποία επιτρέπει την κατασκευή αισθητήρων σε μορφή τσιπ, σε μεγάλη κλίμακα με πολύ μικρό κόστος. Αντίστοιχα με τους αδρανειακούς αισθητήρες, και οι μαγνητικοί αισθητήρες έχουν γίνει τα τελευταία χρόνια αναπόσπαστο κομμάτι πολλών συσκευών της καθημερινής ζωής αλλά και προϊόντων υψηλής τεχνολογίας. Αισθητήρες υψηλής ακρίβειας είναι πλέον διαθέσιμοι σε μορφή τσιπ μαζί με τις απαραίτητες ηλεκτρονικές διατάξεις για τη ψηφιοποίηση και πολλές φορές και τη ψηφιακή επεξεργασία των μετρήσεων κάνοντας εύκολη την ενσωμάτωση των μαγνητικών αισθητήρων σε μεγαλύτερα συστήματα.

Τόσο στην περίπτωση των αδρανειακών αισθητήρων όσο και σε αυτή των μαγνητικών αισθητήρων, κάθε μονάδα αισθητήρα χρειάζεται μια διαδικασία βαθμονόμησης μετά την παραγωγή της προκειμένου να παράγει μετρήσεις υψηλής ακρίβειας. Η διαδικασία αυτή μπορεί να γίνει είτε κατά την κατασκευή του αισθητήρα στο εργοστάσιο, είτε σε μεταγενέστερο χρόνο, πολλές φορές ακόμα και μετά την ενσωμάτωση του αισθητήρα σε ένα μεγαλύτερο σύστημα.

Ένας βασικός περιορισμός που προκύπτει κατά την χρήση αδρανειακών και μαγνητικών αισθητήρων χαμηλού κόστους, είναι η έλλειψη βαθμονόμησης. Σε αυτή την περίπτωση, το κόστος της βαθμονόμησης είναι πολλαπλάσιο του κόστους του αισθητήρα και έτσι είναι ευθύνη του σχεδιαστή του συστήματος να προβλέψει για την βαθμονόμηση του αισθητήρα, όταν αυτή είναι απαραίτητη. Συγκεκριμένα, στην περίπτωση της πλοήγησης, η έρευνα τα τελευταία χρόνια επικεντρώνεται στην χρήση αδρανειακών και μαγνητικών αισθητήρων σε συνεργασία με τα δορυφορικά συστήματα πλοήγησης (GPS, Glonass, Beidou, Galileo) αλλά και στην ανάπτυξη πλήρως αυτόνομων συστημάτων αδρανειακής πλοήγησης. Η πλοήγηση με χρήση αποκλειστικά αδρανειακών αισθητήρων αντιμετωπίζει μεν τα βασικά προβλήματα των δορυφορικών συστημάτων πλοήγησης (δεν λειτουργούν σε εσωτερικούς χώρους, είναι ευαίσθητα σε παρεμβολές, έχουν μικρό ρυθμό ανανέωσης) αλλά απαιτεί

μετρήσεις μεγάλης ακρίβειας και πιστότητας.

Αυτή η εργασία επικεντρώνεται σε αδρανειακούς και μαγνητικούς αισθητήρες τριών αξόνων και χαμηλού κόστους και εξερευνά μεθόδους και αλγόριθμους για την καλύτερη εκμετάλλευση των δεδομένων τους σε εφαρμογές που απαιτούν υψηλή ακρίβεια, όπως για παράδειγμα η πλοήγηση. Στο πρώτο της μέρος παρουσιάζονται μέθοδοι και αλγόριθμοι για την βαθμονόμηση αδρανειακών και μαγνητικών αισθητήρων χωρίς την χρήση ειδικού εργαστηριακού εξοπλισμού. Στη συνέχεια, στο δεύτερο μέρος, αναδεικνύεται η σημασία της βαθμονόμησης σε εφαρμογές πλοήγησης και στη συνέχεια παρουσιάζονται δύο χαρακτηριστικές εφαρμογές.

Μέρος Α: Βαθμονόμηση Αδρανειακών και Μαγνητικών Αισθητήρων

Σε κάθε αισθητήρα, η βαθμονόμηση (calibration) είναι μια απαραίτητη διαδικασία ώστε η μέτρηση του αισθητήρα να ανταποκρίνεται με ακρίβεια στο μετρούμενο μέγεθος. Ο όρος βαθμονόμηση στην παρούσα εργασία χρησιμοποιείται για να εκφράσει όχι μόνο την αντιστοίχιση της κλίμακας της εξόδου του αισθητήρα με αυτή του υπό μέτρηση μεγέθους αλλά και την αντιστάθμιση των στατικών σφαλμάτων του αισθητήρα όπως για παράδειγμα το offset, την μη ορθογωνιότητα των αξόνων κ.α.

Το πρώτο βήμα για την βαθμονόμηση των αδρανειακών και μαγνητικών αισθητήρων είναι η αναγνώριση και η μοντελοποίηση των στατικών πηγών σφάλματος που επηρεάζουν την έξοδο του αισθητήρα. Το στατικό σφάλμα των αισθητήρων οφείλεται κυρίως σε ατέλειες κατά την διαδικασία κατασκευής του αισθητήρα και στις ηλεκτρονικές διατάξεις που χρησιμοποιούνται για την καταγραφή της εξόδου του και είναι στο μεγαλύτερο του μέρος γραμμικό ως προς την έξοδο του αισθητήρα. Έτσι, στην παρούσα εργασία, χρησιμοποιείται ένα γραμμικό μοντέλο για κάθε αισθητήρα που συνδέει την μέτρηση του με το μετρούμενο μέγεθος (επιτάχυνση, γωνιακή ταχύτητα ή μαγνητικό πεδίο) και συμπεριλαμβάνει όλες τις γραμμικές πηγές σφάλματος.

Ιδιαίτερο ενδιαφέρον παρουσιάζει η περίπτωση των μαγνητικών αισθητήρων η έξοδος των οποίων μπορεί να επηρεαστεί και από εξωτερικούς παράγοντες όταν ο αισθητήρας ενσωματωθεί σε ένα μεγαλύτερο σύστημα. Πιο συγκεκριμένα, μαγνητικά υλικά τα οποία μπορεί να υπάρχουν κοντά στον μαγνητικό αισθητήρα, αλλοιώνουν το τοπικό μαγνητικό πεδίο με αποτέλεσμα μια μόνιμη, στατική παραμόρφωση στην έξοδο του αισθητήρα. Το φαινόμενο αυτό συμπεριλαμβάνεται στην μοντελοποίηση και στη συνέχεια αντισταθμίζεται κατά την διαδικασία της βαθμονόμησης. Αφού έχει γίνει η αναγνώριση των πηγών σφάλματος και η μοντελοποίηση της εξόδου του αισθητήρα, η διαδικασία της βαθμονόμησης στοχεύει στον υπολογισμό των διάφορων παραμέτρων του μοντέλου. Δεδομένου ότι η εργασία αυτή στοχεύει κυρίως σε αισθητήρες χαμηλού κόστους, οι αλγόριθμοι βαθμονόμησης που παρουσιάζονται δεν απαιτούν ειδικό εξοπλισμό για την εφαρμογή τους αφού αυτό θα αύξανε σημαντικά το κόστος του αισθητήρα. Στη περίπτωση αυτή, ο αλγόριθμος βαθμονόμησης έχει ακόμα πιο δύσκολο έργο, αφού εκτός από τις παραμέτρους του μοντέλου μέτρησης, άγνωστη είναι και η πραγματική τιμή του μετρούμενου μεγέθους.

Βαθμονόμηση Αισθητήρων Επιτάχυνσης και Μαγνητικού Πεδίου

Οι μετρήσεις των αισθητήρων επιτάχυνσης και μαγνητικού πεδίου μπορούν να περιγραφούν από το ίδιο μαθηματικό μοντέλο. Έτσι, στις περισσότερες περιπτώσεις, οι ίδιοι αλγόριθμοι μπορούν να χρησιμοποιηθούν για τη βαθμονόμηση και των δύο αισθητήρων. Στην βιβλιογραφία, η βαθμονόμηση αισθητήρων επιτάχυνσης και μαγνητικού πεδίου συνήθως εκφράζεται ως ένα πρόβλημα βελτιστοποίησης (optimization) ή εκτίμησης (estimation) από την λύση των οποίων προκύπτουν οι απαραίτητες παράμετροι του μοντέλου μέτρησης. Για την επίλυση του προβλήματος αυτού, όταν δεν χρησιμοποιείται ειδικός εργαστηριακός εξοπλισμός ως αναφορά, χρησιμοποιούνται μια σειρά μετρήσεων της επιτάχυνσης της βαρύτητας ή του μαγνητικού πεδίου της Γης αντίστοιχα.

Τόσο για τα προβλήματα βελτιστοποίησης όσο και για αυτά της εκτίμησης, στη βιβλιογραφία προτείνεται η λύση τους με τη χρήση αριθμητικών μεθόδων. Συγκεκριμένα, στην περίπτωση των προβλημάτων βελτιστοποίησης, η επίλυση τους γίνεται με τη χρήση των μεθόδων κλίσης (gradient descent και Newton-Raphson) ενώ για τα προβλήματα εκτίμησης, συνήθως χρησιμοποιείται κάποια παραλλαγή του φίλτρου Kalman. Και στις δύο περιπτώσεις, παρουσιάζονται τα προβλήματα του αυξημένου υπολογιστικού κόστους αλλά, και της δυσκολίας η και αδυναμίας σύγκλισης των αλγορίθμων. Η δυσκολία στη σύγκλιση οφείλεται σε μεγάλο βαθμό στο γεγονός ότι η επιτυχία των μεθόδων αυτών στηρίζεται σε μεγάλο βαθμό σε μια αρχική εκτίμηση των αγνώστων παραμέτρων. Μάλιστα, πολλές εργασίες προτείνουν διαφορετικές μεθόδους για τον υπολογισμό μιας καλής αρχικής εκτίμησης των αγνώστων με μικρό υπολογιστικό κόστος προκειμένου μετά οι αριθμητικές μέθοδοι να συγκλίνουν με ένα μικρό αριθμό επαναλήψεων.

Στην παρούσα διατριβή χρησιμοποιείται μια διαφορετική προσέγγιση. Το πρόβλημα της βαθμονόμησης διατυπώνεται μεν ως ένα πρόβλημα βελτιστοποίησης, η λύση αυτού όμως δίνεται με μια επαναληπτική μέθοδο η οποία στηρίζεται σε ένα υπολογιστικά ελαφρύ πρόβλημα ελαχίστων τετραγώνων. Η προτεινόμενη μέθοδος, έκτος του πλεονεκτήματος που παρουσιάζει σε όρους υπολογιστικού κόστους, δεν απαιτεί κάποια αρχική εκτίμηση των αγνώστων παραμέτρων με αποτέλεσμα να είναι πιο εύρωστη από αυτές που χρησιμοποιούν τις παραδοσιακές αριθμητικές μεθόδους. Η μεθοδολογία συλλογής των απαραίτητων μετρήσεων είναι σημαντικός παράγοντας για την επιτυχία του αλγορίθμου βαθμονόμησης ο οποίος μάλιστα συχνά παραλείπεται στη βιβλιογραφία. Στην παρούσα εργασία, οι αλγόριθμοι που παρουσιάζονται πλαισιώνονται με μια βήμα-προς-βήμα μεθοδολογία για την συλλογή των απαραίτητων μετρήσεων τόσο για την περίπτωση των αισθητήρων επιτάχυνσης όσο και για αυτή των μαγνητικών αισθητήρων.

Ευθυγράμμιση των Αξόνων Ευαισθησίας Αισθητήρων Επιτάχυνσης και Μαγνητικού Πεδίου

Σε πολλές εφαρμογές, συμπεριλαμβανομένων και αυτών της πλοήγησης, οι μετρήσεις ενός ή περισσότερων αισθητήρων επιτάχυνσης χρησιμοποιούνται σε συνδυασμό με αυτές αντίστοιχων αισθητήρων μαγνητικού πεδίου. Σε αυτές τις περιπτώσεις είναι συνήθως απαραίτητο οι άξονες ευαισθησίας των δύο αισθητήρων να είναι ευθυγραμμισμένοι.

Η έλλειψη ευθυγράμμισης μεταξύ των αξόνων ευαισθησίας μπορεί να οφείλεται σε πολλούς παράγοντες. Αρχικά, όταν δύο ή περισσότεροι αισθητήρες που ενσωματώνονται στο ίδιο σύστημα είναι αναμενόμενο να υπάρξει κάποιο μικρό σφάλμα στην ευθυγράμμιση τους κατά την συναρμο-λόγηση του συστήματος. Επιπλέον, ακόμα και όταν οι αισθητήρες είναι σε ολοκληρωμένη μορφή και αποτελούν κομμάτι του ίδιου τσίπ, ατέλειες κατά την διαδικασία παραγωγής προκαλούν ένα μικρό σφάλμα στην ευθυγράμμιση των αξόνων τους. Τέλος, η χρήση αλγορίθμων βαθμονόμησης οι οποίοι δεν χρησιμοποιούν ειδικό εργαστηριακό εξοπλισμό και δεν στηρίζονται σε κάποιο απόλυτο σύστημα αξόνων αναφοράς (όπως και αυτοί που παρουσιάζονται στην παρούσα εργασία) μπορεί να προκαλέσει περιστροφή των αξόνων ευαισθησίας των δύο αισθητήρων και απώλεια της μεταξύ τους ευθυγράμμισης.

Η μαγνητική έγκλιση (magnetic dip ή magnetic inclination) χρησιμοποιείται συνήθως ως αναφορά για την ευθυγράμμιση των αξόνων αισθητήρων επιτάχυνσης και μαγνητικού πεδίου χωρίς τη χρήση ειδικού εργαστηριακού εξοπλισμού. Η μαγνητική έγκλιση ορίζεται ως η γωνία που σχηματίζει το διάνυσμα του μαγνητικού πεδίου της Γης με το οριζόντιο επίπεδο. Στηριζόμενες στον ορισμό της μαγνητικής έγκλισης και χρησιμοποιώντας ένα πλήθος μετρήσεων, πολλές εργασίες σχηματίζουν ένα πρόβλημα βελτιστοποίησης ή ένα πρόβλημα εκτίμησης παραμέτρων η λύση του οποίου δίνει την σχετική περιστροφή των αξόνων του αισθητήρα επιτάχυνσης και του μαγνητικού αισθητήρα. Ομοίως με την περίπτωση της βαθμονόμησης που περιεγράφηκε προηγουμένως, οι αριθμητικές μέθοδοι κλίσης ή το φίλτρο Kalman χρησιμοποιούνται για την επίλυση του προβλήματος βελτιστοποίησης και εκτίμησης παραμέτρων αντίστοιχα.

Στην εργασία αυτή παρουσιάζεται μία νέα μέθοδος για την ευθυγράμμιση των αξόνων ενός αισθητήρα επιτάχυνσης και ενός αισθητήρα μαγνητικού πεδίου η οποία δίνει μια αναλυτική λύση στο πρόβλημα και δεν στηρίζεται σε κάποια αριθμητική μέθοδο. Η μέθοδος αυτή στηρίζεται στην μαγνητική έγκλιση για τον σχηματισμό ενός προβλήματος βελτιστοποίησης του οποίου η λύση δίνεται σε κλειστή μορφή. Έτσι η προτεινόμενη μέθοδος υπερτερεί της κλασσικής προσέγγισης τόσο σε όρους υπολογιστικού κόστους όσο και σε ευρωστία.

Βαθμονόμηση Γυροσκοπίου

Η βαθμονόμηση του γυροσκοπίου παρουσιάζει ιδιαίτερο ενδιαφέρον τόσο στη διατύπωση ενός μαθηματικού προβλήματος για τον υπολογισμό των απαραίτητων παραμέτρων του μοντέλου μέτρησης όσο και στην εύρεση κατάλληλης μέτρησης αναφοράς. Μία προφανής λύση στο πρόβλημα αυτό θα ήταν να χρησιμοποιηθεί η γνωστή ταχύτητα περιστροφής της γης ως αναφορά για τη διατύπωση και επίλυση ενός προβλήματος βελτιστοποίησης η εκτίμησης παραμέτρων. Ενώ η λύση αυτή μπορεί να χρησιμοποιηθεί σε γυροσκοπία υψηλής ακρίβειας, στην περίπτωση των MEM γυροσκοπίων χαμηλού κόστους η ταχύτητα περιστροφής της Γης είναι αδύνατο να μετρηθεί λόγω του υψηλότερου θορύβου μέτρησης του αισθητήρα.

Η πιο συνήθης λύση που προτείνεται στη βιβλιογραφία για τη βαθμονόμηση του γυροσκοπίου είναι η χρήση ενός αισθητήρα μαγνητικού πεδίου για τον υπολογισμό μιας τιμής αναφοράς για τη γωνιακή ταχύτητα. Στη συνέχεια, ένα πλήθος μετρήσεων σε συνδυασμό με τις τιμές αναφοράς που

έχουν εξαχθεί από τα δεδομένα του μαγνητικού αισθητήρα χρησιμοποιούνται για την επίλυση ενός προβλήματος εκτίμησης παραμέτρων. Όμοια με την περίπτωση της βαθμονόμησης των αισθητήρων επιτάχυνσης και μαγνητικού πεδίου, η προσέγγιση αυτή αντιμετωπίζει στην πράξη προβλήματα σύγκλισης αλλά και χαμηλής ακρίβειας.

Στην παρούσα εργασία προτείνεται μια εντελώς διαφορετική προσέγγιση του προβλήματος. Θεωρώντας ότι όλοι οι αισθητήρες είναι σταθερά τοποθετημένοι σε μία άκαμπτη πλατφόρμα, γίνεται χρήση του αισθητήρα μαγνητικού πεδίου αλλά και του αισθητήρα επιτάχυνσης για τον υπολογισμό των γωνιών περιστροφής μεταξύ έξι στατικών θέσεων της πλατφόρμας. Στη συνέχεια, οι γωνίες αυτές συγκρίνονται με τις αντίστοιχες γωνίες που προκύπτουν από την ολοκλήρωση των μετρήσεων γωνιακής ταχύτητας του γυροσκοπίου κατά την διάρκεια της μετάβασης από την μία στατική θέση στην επόμενη. Οι παράμετροι βαθμονόμησης υπολογίζονται από την επίλυση ενός προβλήματος βελτιστοποίησης το οποίο ελαχιστοποιεί το σφάλμα μεταξύ των γωνιών περιστροφής που υπολογίζονται από το γυροσκόπιο και των αντίστοιχων που υπολογίζονται από τον συνδυασμό του μαγνητικού αισθητήρα και του αισθητήρα επιτάχυνσης.

Παρά το γεγονός ότι το πρόβλημα βελτιστοποίησης λύνεται με την μέθοδο της καθόδου (gradient descent) αλλά και το αυξημένο υπολογιστικό της κόστος λόγω της αριθμητικής ολοκλήρωσης των μετρήσεων του γυροσκοπίου, η προτεινόμενη μέθοδος υπερτερεί σε ευρωστία αλλά και ακρίβεια. Αυτό οφείλεται κυρίως στο γεγονός ότι για τον υπολογισμό της τιμής αναφοράς συμπεριλαμβάνεται και η μέτρηση του αισθητήρα επιτάχυνσης αυξάνοντας τόσο την αξιοπιστία της μέτρησης αναφοράς, όσο και την ακρίβειά της.

Βαθμονόμηση και Ευθυγράμμιση Αξόνων Ευαισθησίας Πολλαπλών Αισθητήρων

Σε πολλές εφαρμογές, περισσότεροι από ένας αδρανειακοί ή μαγνητικοί αισθητήρες χρησιμοποιούνται σε συνδυασμό. Σε αυτές τις περιπτώσεις είναι απαραίτητη τόσο η βαθμονόμηση του κάθε αισθητήρα χωριστά, όσο και η ευθυγράμμιση των αξόνων μεταξύ των αισθητήρων. Μία λύση σε αυτό το πρόβλημα είναι η βαθμονόμηση των αισθητήρων, και η ευθυγράμμιση των αξόνων ευαισθησίας τους να αντιμετωπιστούν σαν δύο διαφορετικές διαδικασίες. Έτσι, θα μπορούσαν να χρησιμοποιηθούν οι αλγόριθμοι που παρουσιάζονται στην παρούσα εργασία για την βαθμονόμηση κάθε αισθητήρα και στη συνέχεια να αναπτυχθεί μία μέθοδος για την ευθυγράμμιση των αξόνων ευαισθησίας τους. Ωστόσο, αυτή η προσέγγιση αυξάνει σημαντικά τόσο το υπολογιστικό κόστος όσο και την πολυπλοκότητα της απαιτούμενης διαδικασίας συλλογής μετρήσεων και βαθμονόμησης.

Στην παρούσα διατριβή παρουσιάζεται μία μέθοδος η οποία αντιμετωπίζει και λύνει το πρόβλημα της βαθμονόμησης και αυτό της ευθυγράμμισης των αξόνων ευαισθησίας πολλαπλών μαγνητικών ή αδρανειακών αισθητήρων σαν ένα ενιαίο πρόβλημα. Σε αυτό το πλαίσιο, το πρόβλημα βελτιστοποίησης που διατυπώθηκε για την περίπτωση ενός μαγνητικού ή αδρανειακού αισθητήρα επεκτείνεται στην περίπτωση πολλαπλών αισθητήρων ενώ σε αυτό πλέον συμπεριλαμβάνεται και το πρόβλημα της ευθυγράμμισης των αξόνων ευαισθησίας των αισθητήρων. Η λύση του νέου προβλήματος βελτιστοποίησης δίνεται και πάλι με την χρήση μιας επαναληπτικής μεθόδου που στηρίζεται στην επίλυση ενός υπολογιστικά φθηνού προβλήματος ελαχίστων τετραγώνων.

Μέρος Β: Εφαρμογές Αδρανειακών και Μαγνητικών Αισθητήρων

Οι αδρανειακοί και οι μαγνητικοί αισθητήρες χρησιμοποιούνται σε ένα μεγάλο εύρος εφαρμογών. Στην παρούσα διατριβή παρουσιάζονται δύο εφαρμογές στον τομέα της πλοήγησης. Αρχικά παρουσιάζεται ένας αλγόριθμος για την πλοήγηση ενός πεζού ανθρώπου ο οποίος αξιοποιεί τον μαγνητικό αισθητήρα για να διορθώσει το μεγάλο σφάλμα στον προσανατολισμό που εισάγεται από το γυροσκόπιο. Στη συνέχεια, παρουσιάζεται μια νέα αρχιτεκτονική για την εκτίμηση της γωνιακής ταχύτητας σε εφαρμογές πλοήγησης. Η αρχιτεκτονική αυτή χρησιμοποιεί ένα γυροσκόπιο και πολλαπλούς αισθητήρες επιτάχυνσης και πετυχαίνει εξαιρετικά χαμηλά επίπεδα θορύβου.

Πλοήγηση Με Χρήση Αδρανειακών και Μαγνητικών Αισθητήρων

Η πλοήγηση με χρήση αδρανειακών αισθητήρων χρησιμοποιείται εδώ και πολλές δεκαετίες, ακόμα και πριν την ανάπτυξη των συστημάτων δορυφορικής πλοήγησης, σε τομείς όπως η ναυτιλία, η αεροπλοΐα και η εξερεύνηση του διαστήματος. Ο κλάδος της αδρανειακής πλοήγησης έχει προσελκύσει έντονο ερευνητικό ενδιαφέρον τα τελευταία χρόνια. Η ανάπτυξη αυτή οφείλεται κυρίως στο πολύ χαμηλό κόστος και την μαζική παραγωγή των αισθητήρων τεχνολογίας MEM, αλλά και στους περιορισμούς που θέτουν οι τεχνολογίες δορυφορικής πλοήγησης (περιορισμένος ρυθμός ανανέωσης, δεν λειτουργούν σε εσωτερικούς χώρους, είναι ευαίσθητες σε παρεμβολές). Μία πολύ δημοφιλής εφαρμογή της αδρανειακής πλοήγησης στην βιβλιογραφία είναι αυτή της πλοήγησης ενός πεζού ανθρώπου. Στην πιο απλή εκδοχή, χρησιμοποιούνται μόνο ένας αισθητήρας επιτάχυνσης και ένα γυροσκόπιο τριών αξόνων οι οποίοι τοποθετούνται στο παπούτσι ενός ανθρώπου. Οι μετρήσεις των δύο αισθητήρων, σε συνδυασμό με τις εξισώσεις της κινηματικής χρησιμοποιούνται για την διατύπωση ενός προβλήματος εκτίμησης παραμέτρων, το οποίο λύνεται με την χρήση του φίλτρου Kalman και δίνει ως αποτέλεσμα μια εκτίμηση της ταχύτητας, του προσανατολισμού και της θέσης.

Στα πλαίσια του φίλτρου Kalman και για την διόρθωση του σφάλματος θέσης που προκύπτει από το σφάλμα μέτρησης των δύο αισθητήρων χρησιμοποιείται και η τεχνική Zero Velocity Update (ZUPT). Η συγκεκριμένη τεχνική βασίζεται στο γεγονός ότι τη στιγμή που το παπούτσι είναι ολόκληρο σε επαφή με το έδαφος έχει μηδενική ταχύτητα. Αρχικά, χρησιμοποιώντας τις μετρήσεις των δύο αισθητήρων (ή και επιπλέον αισθητήρων) γίνεται μία εκτίμηση των χρονικών στιγμών κατά τις οποίες το παπούτσι είναι ολόκληρο σε επαφή με το έδαφος. Στη συνέχεια, οι μετρήσεις των αισθητήρων επιτάχυνσης και γωνιακής ταχύτητας, μαζί με την πληροφορία της μηδενικής ταχύτητας, δίνονται ως είσοδος στο φίλτρο Kalman το οποίο τις χρησιμοποιεί για να αντισταθμίσει το σφάλμα στην εκτίμηση της ταχύτητας, της θέσης και του προσανατολισμού.

Πολλές εργασίες προτείνουν διαφορετικούς αλγορίθμους οι οποίοι χρησιμοποιώντας διαφορετικές διατυπώσεις του προβλήματος, διαφορετικούς αλγορίθμους για την υλοποίηση της τεχνικής ZUPT αλλά και διαφορετικές παραλλαγές του φίλτρου Kalman πετυχαίνουν μεγάλη ακρίβεια στην εκτίμηση της θέσης. Κοινό μειονέκτημα όλων των αλγορίθμων οι οποίοι χρησιμοποιούν αποκλειστικά μετρήσεις αδρανειακών αισθητήρων είναι η σταδιακή μείωση της ακρίβειας τους που μακροπρόθεσμα οδηγεί στην συσσώρευση ενός μεγάλου σφάλματος θέσης. Το σφάλμα αυτό οφείλεται

κυρίως στο σφάλμα μέτρησης αλλά και την σταδιακή μεταβολή του offset του γυροσκοπίου το οποίο οδηγεί σε ένα συσσωρευτικό σφάλμα στον προσανατολισμό και μετέπειτα στην θέση.

Στην παρούσα εργασία παρουσιάζεται ένα σύστημα αδρανειακής πλοήγησης το οποίο συνδυάζει τα δεδομένα αδρανειακών και μαγνητικών αισθητήρων τοποθετημένων στο παπούτσι ενός πεζού ανθρώπου και παρέχει μακροπρόθεσμα μια μεγάλης ακρίβειας εκτίμηση της θέσης. Το προτεινόμενο σύστημα χρησιμοποιεί την τεχνική ZUPT στα πλαίσια ενός προβλήματος εκτίμησης παραμέτρων το οποίο λύνεται με την χρήση του φίλτρου Kalman. Συμπληρωματικά και ανεξάρτητα από την εκτίμηση του φίλτρου Kalman, γίνεται μία δεύτερη εκτίμηση του προσανατολισμού χρησιμοποιώντας τον αισθητήρα επιτάχυνσης και τον αισθητήρα μαγνητικού πεδίου. Σε αντίθεση με παρόμοια συστήματα που προτείνονται στην βιβλιογραφία, η εκτίμηση αυτή δεν απαιτεί την επίλυση κάποιου υπολογιστικά ακριβούς προβλήματος εκτίμησης παραμέτρων ή βελτιστοποίησης. Αντίθετα, γίνεται χρήση του υπολογιστικά φθηνού αλγόριθμου TRIAD κατά τις χρονικές στιγμές κατά τις οποίες ανιχνεύεται μηδενική ταχύτητα. Το σφάλμα της εκτίμησης του προσανατολισμού μέσω του αλγόριθμου TRIAD εξαρτάται μόνο από το στιγμιαίο σφάλμα μέτρησης των αισθητήρων επιτάχυνσης και μαγνητικού πεδίου και δεν είναι συσσωρευτικό όπως αυτό της εκτίμησης του φίλτρου Kalman μέσω των μετρήσεων του γυροσκοπίου. Έτσι, κάθε φορά που είναι διαθέσιμη αυτή η δεύτερη εκτίμηση του προσανατολισμού, χρησιμοποιείται για να αντισταθμίσει το συσσωρευτικό σφάλμα της εκτίμησης του φίλτρου Kalman με αποτέλεσμα μία μεγάλης ακρίβειας εκτίμηση της θέσης σε βάθος χρόνου.

Βελτίωση της Εκτίμησης της Γωνιακής Ταχύτητας με Χρήση Πολλαπλών Αισθητήρων Επιτάχυνσης

Το μεγαλύτερο μέρος του σφάλματος στα αδρανειακά συστήματα πλοήγησης οφείλεται στο συσσωρευτικό σφάλμα μέτρησης του γυροσκοπίου. Στην παρούσα διατριβή, προτείνεται ένα σύστημα κλειστού βρόχου το οποίο χρησιμοποιεί ένα γυροσκόπιο τριών αξόνων και πολλαπλούς αισθητήρες επιτάχυνσης τριών αξόνων για τον υπολογισμό της γωνιακής ταχύτητας με σημαντικά μειωμένο θόρυβο σε σύγκριση με τη μέτρηση του γυροσκοπίου.

Το προτεινόμενο σύστημα στηρίζεται στην ήδη υπάρχουσα θεωρία των συστημάτων πλοήγησης χωρίς χρήση του γυροσκοπίου (gyro-free inertial navigation systems). Η θεωρία αυτή αναπτύχθηκε κατά τα πρώτα χρόνια της εμπορικής διάθεσης αισθητήρων τύπου MEM κατά τα οποία υπήρχαν διαθέσιμοι MEM αισθητήρες επιτάχυνσης αλλά όχι MEM γυροσκόπια. Τοποθετώντας ένα πλήθος αισθητήρων επιτάχυνσης σε ένα στερεό σώμα είναι δυνατός ο υπολογισμός της γωνιακής του ταχύτητας μέσω της επίλυσης ενός συστήματος μη γραμμικών διαφορικών εξισώσεων. Πολλές εργασίες προτείνουν διαφορετικές αρχιτεκτονικές όσον αφορά το πλήθος αλλά και τις θέσεις τοποθέτησης των αισθητήρων επιτάχυνσης με σκοπό την απλοποίηση του συστήματος διαφορικών εξισώσεων (μέσω της απαλοιφής των μη γραμμικών όρων).

Ένα μεγάλο πλεονέκτημα των συστημάτων αυτών είναι το γεγονός ότι εάν οι αισθητήρες επιτάχυνσης τοποθετηθούν σε μεγάλη απόσταση μεταξύ τους, η εκτίμηση της γωνιακής ταχύτητας έχει σημαντικά μειωμένο θόρυβο σε σχέση με την μέτρηση του γυροσκοπίου. Παρόλα αυτά, κοινό μειονέκτημα όλων των προτεινόμενων αρχιτεκτονικών είναι ότι κάθε μικρό σφάλμα τόσο στην μέτρηση των αισθητήρων επιτάχυνσης όσο και στην θέση τοποθέτησης τους πάνω στο στερεό σώμα

μεταφράζεται σε ένα σφάλμα στην εκτιμώμενη γωνιακή ταχύτητα το οποίο κατά την διαδικασία της πλοήγησης μεταφράζεται σε ένα συσσωρευτικό σφάλμα στον προσανατολισμό.

Στην παρούσα εργασία γίνεται χρήση πολλαπλών αισθητήρων επιτάχυνσης για τον υπολογισμό της γωνιακής ταχύτητας, ενώ παράλληλα χρησιμοποιείται και ένα γυροσκοπιο σε ένα σύστημα κλειστού βρόχου για την αντιστάθμιση των σφαλμάτων που προκύπτουν από το σφάλμα μέτρησης των αισθητήρων επιτάχυνσης αλλά και του σφάλματος στην τοποθέτησή τους. Με αυτό τον τρόπο, η έξοδος του προτεινόμενου συστήματος, επηρεάζεται μεν από το σφάλμα μέτρησης του γυροσκοπίου στις πολύ χαμηλές συχνότητες (κατά κύριο λόγο από το offset του γυροσκοπίου), αλλά, στις υψηλότερες συχνότητες, ο θόρυβος της γωνιακής ταχύτητας είναι αισθητά χαμηλότερος σε σύγκριση με αυτόν του γυροσκοπίου προσφέροντας έτσι μια συνολικά πιο ακριβή εκτίμηση της γωνιακής ταχύτητας.

Acknowledgements

First and foremost, I would like to express my appreciation to my supervisor, Professor Pavlos-Petros Sotiriadis, who helped me immensely in the realization of my dissertation. With his scientific expertise he was able to effectively guide me and help me to overcome several roadblocks and obstacles. It has been an honor to have worked under his supervision and to have received his support.

I would like also to deeply thank Professor Athanasios Panagopoulos and Professor Emeritus Kiamal Pekmestzi for serving as members of my advisory committee. In addition, I would like to thank Professor Evangelos Hristoforou, Professor Evangelos Papadopoulos, Professor Panayiotis Psarrakos and Professor Ioannis Papananos for participating in the seven-member examination committee.

I would like to wholeheartedly thank all my colleagues in the Circuits and Systems Group, with whom I shared several moments during the past six years. Our collaboration and friendship were crucial in making the PhD experience as pleasurable as possible.

Lastly, I would like to thank my family and friends for their love, support and encouragement during all the years of my studies.

This research is co-financed by Greece and the European Union (European Social Fund-ESF) through the Operational Programme «Human Resources Development, Education and Lifelong Learning» in the context of the project “Strengthening Human Resources Research Potential via Doctorate Research” (MIS-5000432), implemented by the State Scholarships Foundation (IKY)

List of Publications

International Peer-Reviewed Journals

1. **K. Papafotis**, P.P. Sotiriadis, "MAG.I.C.AL. – A Unified Methodology for Magnetic and Inertial Sensors Calibration and Alignment", *IEEE Sensors Journal*, vol. 19, no. 18, pp. 8241-8251, 15 Sept.15, 2019.
2. **K. Papafotis**, P.P. Sotiriadis, "Multiple Accelerometers and Magnetometers Joint Calibration and Alignment", *IEEE Sensors Letters*, Vol. 4, Issue 3, March 2020, pp. 2475-1472.
3. **K. Papafotis**, P.P. Sotiriadis, "A Fast and Accurate Accelerometer and Magnetometer Alignment Algorithm", *IEEE Sensors Journal*, vol. 20, no. 24, pp. 15061-15067, 15 Dec.15, 2020.
4. P.P. Sotiriadis, **K. Papafotis**, "A Single-Step Method for Accelerometer and Magnetometer Axes Alignment", *IEEE Trans. on Instrumentation and Measurement*, vol. 70, pp. 1-7, 2021.
5. P.P. Sotiriadis, **K. Papafotis**, "Accurate Analytical Accelerometer–Magnetometer Axes Alignment Guaranteeing Exact Orthogonality," *IEEE Trans. on Instrumentation and Measurement*, vol. 70, pp. 1-7, 2021.
6. **K. Papafotis**, D. Nikitas, P.P. Sotiriadis, "Magnetic Field Sensors' Calibration: Algorithms' Overview and Comparison", *Sensors* 2021, 21, 5288.
7. N. Hadjigeorgiou, K. Asimakopoulos, **K. Papafotis**, P.P. Sotiriadis, "Vector Magnetic Field Sensors: Operating Principles, Calibration and Applications," *IEEE Sensors Journal*, vol. 21, no. 11, pp. 12531-12544, 1 June1, 2021.
8. **K. Papafotis**, P.P. Sotiriadis, "Accelerometer and Magnetometer Joint Calibration and Axes Alignment", *Technologies* 2020, 8, 11. **[Invited]**
9. N. Hadjigeorgiou, **K. Papafotis**, P.P. Sotiriadis, "Exploring AMR Sensor's Performance Limits Using an Agile, High-Speed Set-Reset Pulse Generation Circuit", *IEEE Trans. on Magnetics*, vol. 56, no. 9, pp. 1-10, Sept. 2020.

10. N. Voudoukis, C. Dimas, K. Asimakopoulos, D. Baxevanakis, **K. Papafotis**, K. Oustoglou, P.P. Sotiriadis, "The Importance of Introducing the OCTC Method to Undergraduate Students as a Tool for Circuit Analysis and Amplifier Design", *Technologies* 2020, 8, 7. **[Invited]**

International Peer-Reviewed Conferences

1. **K. Papafotis**, P.P. Sotiriadis, "Exploring the Importance of Sensors' Calibration in Inertial Navigation Systems ", *IEEE International Symposium on Circuits and Systems*, Seville, Spain, 2020.
2. **K. Papafotis**, Paul P. Sotiriadis, "Computationally Efficient Calibration Algorithm for Three-Axis Accelerometer and Magnetometer", *IEEE International Conference on Modern Circuits and Systems Technologies*, Thessaloniki, Greece, 2019. **[Best Paper Award]**
3. Dimitris Nikitas, **K.Papafotis** and Paul P. Sotiriadis, "Improving Gyroscope's Noise Performance By Embedding it in a Closed-Loop Involving Multiple Accelerometers", *11th International Conference on Circuits and Systems Technologies (MOCASST)*, Bremen, Germany, 2022
4. D. Nikitas, **K. Papafotis**, P.P. Sotiriadis, "Time-Near-Optimal Longitudinal Control for Quadrotor UAVs", *IEEE International Conference on Modern Circuits and Systems Technologies*, Thessaloniki, Greece, 2021.
5. N. Voudoukis, D. Baxevanakis, **K. Papafotis**, C. Dimas, C. Oustoglou, P. Sotiriadis, "Introducing Senior Undergraduate Students to the Open-Circuit Time-Constant Method for Circuit Analysis", *IEEE International Conference on Modern Circuits and Systems Technologies*, Thessaloniki, Greece, 2019.

Table of Contents

1	Introduction	33
1.1	Motivation and Scope	33
1.2	Thesis Outline	35
2	Inertial & Magnetic Field Sensors' Calibration & Axes Alignment	39
2.1	Introduction	39
2.2	Measurement Model of Inertial and Magnetic Field Sensors	41
2.2.1	Accelerometer's Measurement Model	41
2.2.2	Magnetometer's Measurement Model	41
2.2.3	Gyroscope's Measurement Model	42
2.3	Magnetometer's Calibration	43
2.4	Accelerometer's Calibration	45
2.5	Performance Evaluation of Accelerometer's and Magnetometer's Calibration Algorithm	47
2.5.1	Algorithm's Convergence	47
2.5.2	Measuring Distance Between Calibration Parameter Sets	48
2.5.3	The Effect of Calibration on Sensors' Measurements	50
2.6	Accelerometer's and Magnetometer's Axes Alignment	51
2.6.1	Problem Statement	52
2.6.2	Prior Art and its Limitations	53
2.6.3	The Proposed Algorithm	55
2.6.4	Finding an Initial Condition	59
2.6.5	Evaluation of the Proposed Method	62
2.7	Gyroscope's Calibration	65
2.7.1	Rotation From Accelerometer and Magnetometer Data	65
2.7.2	Rotation From Gyroscope Data	66
2.7.3	Calibration Algorithm	66
2.7.4	Algorithm's Evaluation	68
3	Magnetic Field Sensors' Calibration: Algorithms' Overview and Comparison	73
3.1	Introduction	73
3.2	Magnetic Field Sensor's Error Sources and Measurement Model	74

3.3	Alonso and Shuster (TWOSTEP) [2]	76
3.3.1	Initial Estimate	77
3.3.2	Solution Improvement Step	78
3.4	Crassidis et al. [5]	80
3.5	Dorveaux et al. [7]	81
3.6	Vasconcelos et al. [30]	83
3.6.1	Initial Estimate	84
3.7	Ali et al. [1]	85
3.8	Wu and Shi [31]	87
3.8.1	Initial Estimate	88
3.9	Papafotis and Sotiriadis (MAG.I.C.AL.) [25]	89
3.10	Algorithms' Evaluation and Comparison	91
3.10.1	Synthetic Data Generation	91
3.10.2	Algorithms' Evaluation Using Real Data	105
3.11	Conclusion	106
3.12	Appendix	108
3.12.1	Gradient Vector and Hessian Matrix for [30]	108
3.12.2	Gradient Vector and Hessian Matrix for [31]	109
4	Multiple Accelerometers and Magnetometers Joint Calibration and Alignment	113
4.1	Introduction	113
4.2	Sensors' Measurement Model and The Problem of Calibration	114
4.2.1	Accelerometer's Measurement Model	115
4.2.2	Magnetometer's Measurement Model	115
4.2.3	Calibration as an Optimization Problem	115
4.3	The Proposed Algorithm	117
4.3.1	Algorithm Description	117
4.3.2	Measurements Acquisition Procedure	118
4.4	Experimental and Simulation Results	119
5	Hardware Design of a Low-Cost IMU	125
6	The Importance of Sensors' Calibration in Inertial Navigation Systems	127
6.1	Introduction	127
6.2	Inertial Sensors' Error Propagation In The Inertial Frame	129
6.2.1	Notation	129
6.2.2	Attitude Error Propagation	129
6.2.3	Velocity Error Propagation	130
6.3	Experimental Results	131
6.3.1	Pedestrian Inertial Navigation Using Shoe-Mounted Inertial Sensors	131
6.3.2	Experiment Procedure	131

6.3.3	Error Propagation	131
6.3.4	Trajectory reconstruction	132
7	MAG.I.NAV.	137
7.1	Introduction	137
7.2	The Proposed Inertial Navigation System	139
7.2.1	Notation and Assumptions	139
7.2.2	Top-Level System Architecture	140
7.2.3	Zero Velocity Detection	141
7.2.4	Magnetic Disturbance Detection	142
7.2.5	Attitude Estimation Using Gyroscope's Measurements	143
7.2.6	Attitude Estimation via TRIAD Algorithm Using Accelerometer's and Magnetometer's Measurements	143
7.2.7	Output Attitude Derivation	144
7.2.8	Velocity and Position Estimation Via Kinematic Equations Using Accelerometer's Measurements	145
7.2.9	Zero Velocity Update Method	146
7.2.10	System's Initialization	148
7.2.11	Algorithmic Implementation	148
7.3	Experimental Results	150
8	A Class of Low-Noise Inertial Measurement Units	157
8.1	Introduction	157
8.2	Gyroscope-Free Inertial Measurement Units	158
8.2.1	Principle of Operation	158
8.2.2	Existing Art and Performance Limitations	160
8.3	The Proposed System	160
8.3.1	System Architecture	160
8.3.2	The linear case	161
8.3.3	The General, Non-Linear Case	163
8.3.4	Output Noise Modeling	167
8.4	Results	168
8.4.1	Noise Model Accuracy	169
8.4.2	Noise Performance	170
8.4.3	System Design and Stability Considerations	170
9	Conclusions and Further Research	177
10	Appendix A: IMU Schematics	179

List of Figures

2.1	Convergence of accelerometer calibration algorithm	47
2.2	Convergence of magnetometer calibration algorithm	48
2.3	Normalized magnitude of raw and calibrated accelerometer measurements in six still orientations.	50
2.4	Normalized magnitude of raw and calibrated magnetometer measurements in five still orientations.	50
2.5	Axes alignment.	51
2.6	Magnetic inclination.	51
2.7	Convergence of Newton-Raphson for different values of the weighting factor, λ (Fastest convergence corresponds to $\lambda = 10$).	54
2.8	Distance of R from orthogonality $D_O(R)$ when Newton-Raphson has converged, and, number of iterations required for convergence, as functions of weighting factor λ	54
2.9	Convergence of Newton-Raphson for different values of the weighting factor, λ , when the single-step method in [30] is used for initialization.	55
2.10	Distance of R from orthogonality and Newton-Raphson iterations until convergence for different values of the weighting factor, λ , when the single-step method in [30] is used for initialization.	55
2.11	Convergence of gyroscope calibration algorithm	68
3.1	Two synthetic datasets generated using Algorithm 12 for $K = 150$ (a) and $K = 300$ (b) respectively.	93
3.2	Performance characteristics of the presented algorithms for different values of γ	98
3.3	Performance characteristics of the presented algorithms for different values of β	100
3.4	Performance characteristics of the presented algorithms for different values of K	102
3.5	Performance characteristics of the presented algorithms for different values of the noise variance σ^2	104
3.6	Performance characteristics of the presented algorithms using multiple datasets of real data from two different commercial magnetic field sensors.	106
4.1	Algorithm convergence using measured data - Fractional mean magnitude error.	119

4.2	Algorithm convergence using simulated data.	120
4.3	Raw and calibrated magnetometer's measurements magnitude in different orientations.	120
5.1	System-level diagram of the designed IMU.	125
5.2	The designed IMU printed circuit board (up) and enclosure (down).	126
6.1	Evolution of attitude error in time.	132
6.2	Evolution of velocity error in time.	132
6.3	Reconstructed trajectory using a) uncalibrated inertial sensors, b) calibrated accelerometer and offset compensated gyroscope and c) fully calibrated inertial sensors	133
7.1	Architecture of the proposed pedestrian inertial navigation system.	141
7.2	Qualitative representation of the attitude error when using the proposed attitude estimation scheme. The three operation states are denoted as S1, S2 and S3 respectively.	145
7.3	Walk path inside the campus of the National Technical University of Athens, Greece.	150
7.4	Reconstructed walk path using accelerometer's and gyroscope's measurements and the zero velocity update method.	151
7.5	Reconstructed walk path using the measurements of both inertial and magnetic sensors, the zero velocity update method and the proposed attitude correction scheme.	151
7.6	Reconstructed walk path with highlighted some important accuracy characteristics.	152
7.7	Reconstructed walk path using the measurements of both inertial and magnetic sensors, the zero velocity update method and the proposed attitude correction scheme.	153
8.1	Top-level architecture of the proposed system.	161
8.2	Block diagram representation of the proposed system for specific accelerometers' configurations eliminating the non-linear terms of (8.7).	161
8.3	Block diagram representation of the proposed system linearized around $\omega = 0$ rad/s.	167
8.4	IMU configuration composed of nine single-axis accelerometers (grouped in three three-axis ones) and a single three-axis gyroscope.	169
8.5	The PSD of the output of the proposed system (x-axis) compared to the PSD calculated using the noise model of (8.49).	169
8.6	The PSD of the proposed system's output noise (X-axis) compared to the PSD of the gyroscope's output noise (X-axis) for $l = 1m$ (a) and $l = 4m$ (b).	170

List of Tables

2.1	Mean value and variance of the error ε of the proposed method, a gradiend descent (GD) based method, a Newton-Raphson (NR) based method, a Newton-Raphson based method initialized using the solution of [30] and the single-step method of [30] alone.	63
2.2	Performance characteristics of the accelerometer (A) and the magnetometer (M) included in the designed measurement device.	63
2.3	Residual error of the proposed method, a gradiend descent (GD) based method, a Newton-Raphson (NR) based method, a Newton-Raphson based method initialized using the solution of [30] and the "single-step" method of [30] evaluated using five different datasets (D1-D5) of real sensors' data.	64
2.4	Measured roll angle for 90° rotation about x-axis.	69
3.1	Notation	75
3.2	Baseline Evaluation of the presented algorithms.	97
3.3	Operation parameters of the two magnetic field sensors.	105
3.4	Algorithms' Comparison Summary	108
5.1	Basic performance characteristics of the LSM9DS1 SiP (Accelerometer).	126
5.2	Basic performance characteristics of the LSM9DS1 SiP (Gyroscope).	126
5.3	Basic performance characteristics of the LSM9DS1 SiP (Magnetometer).	126
7.1	Notation	140
7.2	Operation states.	144
7.3	Basic performance characteristics of the accelerometer (A), gyroscope (G) and magnetometer (M) included in the LSM9DS1 SiP.	150
7.4	Mean execution time of the proposed algorithm compared to the mean execution time of the algorithms based on Madgwick's AHRS and Mahony's AHRS.	153
8.1	Angular velocity noise within $[0, 100]Hz$ for $\omega_{max} = 1rad/s$ and different values of f_p and g	171
8.2	Angular velocity noise within $[0, 100]Hz$ for $\omega_{max} = 5rad/s$ and different values of f_p and g	171

- 8.3 Angular velocity noise within $[0, 100]Hz$ for $\omega_{max} = 7rad/s$ and different values of f_p and g 171
- 8.4 Angular velocity noise within $[0, 100]Hz$ for $\omega_{max} = 10rad/s$ and different values of f_p and g 172

List of Algorithms

1	Magnetometer Calibration Algorithm	44
2	Accelerometer Calibration	46
3	Proposed Method	62
4	Gyroscope Calibration	67
5	Alonso and Shuster (TWOSTEP) [2]	79
6	Crassidis et al. (Kalman Filter) [5]	81
7	Dorveaux et al. [7]	83
8	Vasconcelos et al. [30]	85
9	Ali et al. (PSO calibration) [1]	87
10	Wu and Shi [31]	89
11	Papafotis and Sotiriadis (MAG.I.C.AL.) [25]	90
12	Generation of Synthetic Data	92
13	Proposed Algorithm	118
14	Proposed Navigation Algorithm.	149

Abbreviations

AI	Artificial Intelligence
AHRS	Attitude and Heading Reference System
BIBO	Bounded-Input, Bounded-Output
EKF	Extended Kalman Filter
GD	Gradient Descent
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LPV	Linear Parameter-Varying
MEMS	Micro-Electro-Mechanical Systems
PSD	Power Spectral Density
PSO	Particle Swarm Optimization
RF	Radio-Frequency
SiP	System in Package
ZUPT	Zero Velocity Potential Update

1

Introduction

1.1 Motivation and Scope

Inertial sensors (accelerometers and gyroscopes) along with magnetic field sensors are nowadays used in a great variety electronic devices. Their application span extends from commercial devices such as smartphones, activity trackers etc. to high-end, industrial, marine, aerospace and military ones. Some representative applications include, but not limited to, navigation, attitude estimation, image stabilization, healthcare systems, gaming and entertainment devices, space exploration and many other industrial and commercial ones.

A very important aspect, especially in high-end applications where measurement accuracy is of major importance, is the sensors' calibration. For every measurement instrument or sensor, calibration ensures that each measurement corresponds to the actual value of the measured quantity and it is not affected by other static or dynamic parameters. The simplest form of calibration of a sensor is to properly scale the its output to match the unit system of the measured quantity. However, when accuracy is important, a calibration procedure must also compensate for other measurement distortions, mainly caused by imperfections during the manufacturing procedure of the sensor.

In many high-end applications, expensive, factory calibrated sensors are used in order to achieve high accuracy. In such cases, very accurate rate tables and magnetic field sources are used as reference to calibrate inertial and magnetic field sensors respectively. However, in applications where cost is important, such as commercial electronic devices, low-cost sensors, typically in integrated form, are preferred. In this case, factory calibration or other calibration services as well as expensive calibration equipment cannot be used as both would raise the sensor's cost significantly.

The main problem when calibration without using special equipment is concerned, is to find a proper measurement reference. To this end, most existing works use the Earth's gravity and magnetic field as reference in order to calibrate inertial and magnetic field vectors respectively when no special equipment is available. Using several sensors' measurements they formulate and solve either an optimization problem or an estimation one to derive the required calibration

parameters.

In addition to the calibration of a single sensor, in several applications such as navigation and heading estimation, the data of the three sensors are combined and thus it is required for their sensitivity axes to be aligned. To this purpose, many works use the magnetic inclination as reference and given several accelerometer's and magnetometer's measurements, they derive the rotation between the sensitivity axes of the two sensors. In addition, several works propose calibration algorithms that exploit the Earth's gravity and local magnetic field in order to align the sensitivity axes of a gyroscope with the ones of an accelerometer or a magnetometer respectively.

While there are several works in the literature proposing different algorithms for inertial and magnetic field sensors' calibration, most of them come with two great disadvantages: a) they do not consider the calibration of inertial and magnetic field sensors as a single problem and deal only with the calibration of a single sensor and b) they use numerical tools to derive the required calibration parameters resulting in increased computational burden and potential convergence issues.

Here, it should be noticed that a calibration procedure, in its strict definition, would require to derive the scale and sensitivity of the sensor, using an absolute reference and a well defined coordinate frame. Thus a complete calibration of inertial and magnetic field sensors without using any piece of equipment and an external reference is not possible. However, existing works on the field use the term calibration to indicate a statistical analysis of the sensors' measurements that compensates for the most important measurement errors and leads to high-accuracy differential measurements. In this work we adopt this, more relaxed, use of the term calibration.

In the first part of this thesis, a complete methodology for calibrating three-axis inertial and magnetic field sensors is introduced. First an algorithm for calibrating three-axis accelerometers and three-axis magnetometers is presented. Using several measurements of the gravity acceleration and the local magnetic field respectively, the calibration parameters are derived as the solution of an optimization problem. Unlike the existing literature, the solution of this problem is not derived using the standard descent methods, but instead, a novel iterative algorithm based on the solution of a computationally cheap least squares problem is introduced.

Then, the problem of aligning the sensitivity axes of the two sensors is examined and formulated as an optimization problem for deriving a rotation matrix representing the rotation between the sensitivity axes of the two sensors. Using several measurements of the two sensors, and exploiting the magnetic inclination phenomenon, the required rotation matrix is derived in closed-form.

Finally, using the calibrated data of the two sensors, a new approach for calibrating the gyroscope is introduced. More specifically, assuming that all three sensors are mounted on the same rigid platform, the rotation of the platform between two still positions calculated using the accelerometer's and the magnetometer's measurements is used as reference to calibrate the gyroscope. The calibration is expressed as an optimization problem which is solved using the gradient descent method.

In the second part, two popular applications of inertial sensors in the field of navigation are presented. Inertial navigation systems use inertial sensors (accelerometers and gyroscopes) to calculate the velocity, orientation and position of a moving object. They are widely used both alone and in combination with satellite-based systems in a wide range of applications including aircraft navigation, spacecrafts, submarines and ships as they provide accurate position estimation and require no extra infrastructure.

The most basic inertial navigation system is comprised of a three-axis accelerometer and a three-axis gyroscope. In theory, using their measurements and applying the standard kinematic equations one can derive the velocity, orientation and position of a moving object. However, in practice, the sensors' measurement errors and noise make inertial navigation a challenging task. Existing works propose different architectures for fusing the measurements of the two sensors in an optimal way, most commonly using a Kalman filter in order to eliminate the effects of the measurement errors and noise. In addition, several works propose architectures including more sensors such as magnetic field sensors, cameras and others in order to further improve the navigation accuracy.

In this thesis, a novel system for pedestrian navigation using inertial and magnetic field sensors is introduced. The proposed system uses a three-axis accelerometer, a three-axis gyroscope and a three-axis magnetometer in order to compensate for the gyroscope's bias drift and provide long-term accurate velocity, orientation and position estimates. To do so, a popular pedestrian navigation approach using the accelerometer's and the gyroscope's measurements in a Kalman filter estimation problem is exploited. In addition, a more accurate, independent, estimate of the orientation is derived using the accelerometer and the magnetometer. A long term accurate orientation estimate is derived by combining the Kalman filter's orientation estimate with the one derived using the accelerometer and the magnetometer.

Then, to further reduce the effect of the gyroscope's noise, an inertial measurement unit architecture utilizing several three-axis accelerometers and a single three-axis gyroscope is presented. The proposed architecture is an extension of gyroscope-free inertial measurement units, which use only accelerometers to measure the linear acceleration and translate it to angular velocity by solving a system of nonlinear differential equations. More specifically, it exploits the basic operating principles of the gyroscope-free inertial measurement systems and uses a three-axis gyroscope to correct the accelerometers' offset. It requires a minimum of three accelerometers which can be placed arbitrarily on a rigid object.

1.2 Thesis Outline

This thesis is divided in two parts. In the first part a complete methodology for calibrating inertial and magnetic field sensors is introduced. Then, in the second part, two applications of inertial sensors in the field of navigation are presented. More specifically, the rest of this thesis is organized as follows.

In Chapter 2, a complete methodology for calibrating and aligning the sensitivity axes of

inertial and magnetic field sensors is introduced. More specifically, the problem of calibrating a three-axis accelerometer, a three-axis magnetometer and a three-axis gyroscope is examined and a calibration algorithm for each sensor is presented and analyzed in detail. In addition, a methodology for aligning the sensitivity axes of an accelerometer and a magnetometer is introduced.

In Chapter 3, the calibration of three-axis magnetic field sensors is reviewed. Seven representative algorithms dealing with the in-situ calibration of magnetic field sensors without requiring any special piece of equipment are reviewed. The algorithms are presented in a user friendly, directly applicable step-by-step form, and are compared in terms of accuracy, computational efficiency, and robustness using both real sensors' data and artificial data with a known sensor measurement distortion.

In Chapter 4, the presented calibration methodology is extended the case where multiple accelerometers or magnetometers are used together. More specifically, a computationally efficient algorithm for simultaneous joint calibration and axes alignment of multiple 3-axis accelerometers or 3-axis magnetometers is presented.

In Chapter 5, the hardware design of a low-cost inertial measurement unit is presented.

In Chapter 6 the importance of sensor's calibration when they are used in inertial navigation systems is explored. First, a bound for the velocity and orientation error as a function of the accelerometer's and gyroscope's calibration parameters is derived. Then, using an inertial measurement unit and a popular pedestrian navigation algorithm, it is experimentally demonstrated how the large error characteristics of low-cost sensors significantly affect the navigation's accuracy.

In Chapter 7, a pedestrian navigation system using inertial and magnetic field sensors is introduced. The proposed system architecture is presented and analyzed in detail while extensive experiments prove its long-term accuracy compared to existing works using only inertial sensors.

In Chapter 8 a new inertial measurement unit architecture is introduced which gives an estimation of the angular velocity with significantly lower noise than the gyroscope's measurement. The proposed system is presented and analyzed in detail while a series of simulations are presented to demonstrate its performance.

Finally, Chapter 9 concludes the thesis with a general summary of its contributions and a presentation of open problems, paving the way for future work.

Part 1

Inertial and Magnetic Field Sensors Calibration

2

Inertial & Magnetic Field Sensors' Calibration & Axes Alignment

2.1 Introduction

Inertial sensors, accelerometers and gyroscopes, are combined with magnetometers in a wide range of applications. For example, in [18] and [31] the three sensors are combined in pedestrian navigation applications. Two heading estimation algorithms based on their joint data are presented in [16] and [20]. Many applications where the three sensors are combined can be found in the literature, from low-cost commercial systems to high-accuracy marine, aerospace and military systems.

Especially in the case of low-cost systems, micro-electro-mechanical (MEMS) inertial sensors are usually preferred due to their significantly lower cost and small size. However, a major disadvantage of MEMS inertial sensors is their large error characteristics [10]. So, in order to use them in applications where accuracy is important, such as navigation, a calibration procedure that compensates for the deterministic part of their error is required.

Similarly to the inertial MEMS sensors, low-cost magnetometers also suffer from significant measurement errors. Apart from the sensor's manufacturing imperfections, the measured magnetic field is strongly distorted by nearby magnetic materials. Surrounding electronic components and the sensor's enclosure are a common source of such distortions. Getting an accurate magnetic field measurement requires a calibration procedure to compensate for both sensor's measurement error and the distortions caused by nearby objects.

Copyright © IEEE. Chapter 2 is reprinted, with permission, from: a)K. Papafotis, P.P. Sotiriadis, "MAG.I.C.AL. – A Unified Methodology for Magnetic and Inertial Sensors Calibration and Alignment", IEEE Sensors Journal, vol. 19, no. 18, pp. 8241-8251, 15 Sept.15, 2019, b)P.P. Sotiriadis, K. Papafotis, "Accurate Analytical Accelerometer-Magnetometer Axes Alignment Guaranteeing Exact Orthogonality," IEEE Trans. on Instrumentation and Measurement, vol. 70, pp. 1-7, 2021., c)P.P. Sotiriadis, K. Papafotis, "A Single-Step Method for Accelerometer and Magnetometer Axes Alignment", IEEE Trans. on Instrumentation and Measurement, vol. 70, pp. 1-7, 2021. and d) K. Papafotis, P.P. Sotiriadis, "Multiple Accelerometers and Magnetometers Joint Calibration and Alignment", IEEE Sensors Letters, Vol. 4, Issue 3, March 2020, pp. 2475-1472. Personal use of this material is permitted, but republication/redistribution requires IEEE permission.

In the case of low-cost inertial and magnetic sensors, factory calibration or after-production calibration using expensive equipment is not an option as it would raise the sensor's cost significantly. Thus, a calibration method that is not based on any external equipment is highly preferred.

For 3-axis accelerometer calibration, most authors take advantage of the fact that the measured magnitude of the specific force is constant when the sensor is still, independently of its orientation. The work in [19] proposes an off-line calibration method based on maximum likelihood estimation. In [2] an algorithm based on least-square method is proposed. In [42] the authors propose a solution based on the Levenberg-Marquardt algorithm to improve the calibration accuracy. The authors in [9] use a nonlinear parameter estimator based on the unscented transformation to calculate the calibration parameters. In [7], calibration parameters are calculated by solving a nonlinear optimization problem.

Gyroscope calibration is a more complicated problem as no convenient rotation reference is available. Some authors use special equipment in order to calibrate a gyroscope [34] [22] [41]. In [33], the authors use the earth's rotation as reference, an approach suffering from the MEMS gyroscope relatively high noise levels. In [26] the rotation of a calibrated accelerometer is used as a reference in a least squares problem formulation. Authors in [37] use a calibrated magnetometer in a Kalman filter estimation problem to calculate the calibration parameters.

For 3-axis magnetometer calibration, the magnetic field of the earth is most commonly used as reference. In [1] , [38], [14], [13] and [35] the authors derive the calibration parameters by solving a maximum likelihood estimation problem. A least-squares based iterative algorithm for magnetometer calibration is proposed in [5]. In [39], the authors formulate the magnetometer calibration as a state estimation problem which can be solved using Kalman filtering.

In most navigation or heading estimation applications the measurements of the three (accelerometer, gyroscope, magnetometer) are combined to give a more accurate result. This gives rise to the need of alignment between the axes of the three sensors. In [12] and [8], magnetometer's axes are aligned with those of the accelerometer. The authors in [36] and [39] use a gyroscope to align magnetometer and inertial sensors. An algorithm for calibration and axes alignment between a gyroscope and an accelerometer is proposed in [40].

The calibration of inertial and magnetic sensors, and the alignment of their axes is required in all relevant applications. However, most of the previous works deal only with the calibration of either a single sensor or the alignment between a pair of them.

This work introduces a complete methodology for unified calibration and joint axes alignment of 3-axis magnetometer, 3-axis accelerometer and 3-axis gyroscope. The proposed method compensates for all linear time-invariant distortions such as scale-factor, cross-coupling and offset, including the soft-iron and hard-iron distortions of the magnetometer. It introduces a new, computationally efficient, least-squares based, iterative algorithm for the calibration of the magnetometer and the accelerometer. It solves the axes alignment problem in an analytical while it also introduces a new way to calibrate the gyroscope based on the sensors' joint data. Finally, a 15-step calibration procedure requiring no external piece of equipment and no external attitude references is introduced.

2.2 Measurement Model of Inertial and Magnetic Field Sensors

Before presenting the calibration algorithms, it is important to introduce the measurement models of the three sensors on which the algorithms are based. Note that in this work, we only consider the most important linear, time invariant error sources for all three sensors to derive the corresponding measurement models.

2.2.1 Accelerometer's Measurement Model

Accelerometer's measurement is modeled as [10] [23]

$$y_a = f + T_{sf}f + T_{cc}f + h_a + \varepsilon, \quad (2.1)$$

where

y_a : 3×1 measurement vector

f : 3×1 true specific force vector

T_{sf} : 3×3 diagonal matrix representing the scale-factor error

T_{cc} : 3×3 matrix representing the cross-coupling error

h_a : 3×1 accelerometer's bias vector

ε : random error

Defining $T_a \triangleq I_3 + T_{sf} + T_{cc}$, where I_3 is the 3×3 identity matrix, (2.1) can be written as

$$y_a = T_a f + h_a + \varepsilon \quad (2.2)$$

2.2.2 Magnetometer's Measurement Model

A magnetometer measures the strength and the direction of the local magnetic field. The measured field is a combination of the earth's magnetic field and an additive field created by magnetic objects attached to the same reference frame as the sensor. This additive field is called hard-iron distortion and causes a permanent bias in the sensor's output.

In addition, magnetometer's measurement is distorted by nearby materials attached to the sensor's frame that influence the magnetic field but don't generate a magnetic field themselves, most commonly metals. This type of distortion is called soft-iron distortion, and, along with the hard-iron distortion are the most important error contributors in the measurements.

Taking into account the hard-iron and soft-iron distortion which are the two dominant sources of distortion, the sensor's measurement can be modeled as [35], [12], [38], [21]

$$y_m = T_{sf}T_{cc}(T_{si}m + h_{hi}) + h_b + \varepsilon \quad (2.3)$$

where

- y_m : 3×1 measurement vector
- m : 3×1 true magnetic field vector
- T_{sf} : 3×3 diagonal matrix representing the scale-factor error
- T_{cc} : 3×3 matrix representing the cross-coupling error
- T_{si} : 3×3 matrix representing the soft-iron distortion
- h_b : 3×1 magnetometer's bias vector
- h_{hi} : 3×1 bias vector due to hard-iron distortion
- ε : random error

Setting $T_m \triangleq T_{sf}T_{cc}T_{si}$ and $h_m \triangleq T_{sf}T_{cc}h_{si} + h_b$, the magnetometer's measurement model becomes

$$y_m = T_m m + h_m + \varepsilon \quad (2.4)$$

2.2.3 Gyroscope's Measurement Model

Gyroscope's measurement is modeled as [10], [23]

$$y_g = \omega + T_{sf}\omega + T_{cc}\omega + h_g + \varepsilon, \quad (2.5)$$

where

- y_g : 3×1 measurement vector
- ω : 3×1 true angular velocity vector
- T_{sf} : 3×3 diagonal matrix representing the scale-factor error
- T_{cc} : 3×3 matrix representing the cross-coupling error
- h_g : 3×1 gyroscope's bias vector
- ε : random error

Defining $T_g = I_3 + T_{sf} + T_{cc}$, (2.5) can be written as

$$y_g = T_g \omega + h_g + \varepsilon \quad (2.6)$$

2.3 Magnetometer's Calibration

The 3-axis magnetometer's calibration is based on the fact that the measured magnitude of the magnetic field should be independent of the magnetometer's orientation. This is formulated as an optimization problem which is solved using a novel least-squares based iterative algorithm achieving fast convergence and computational efficiency.

The purpose of the calibration algorithm is to estimate the calibration parameters T_m and h_m in order to minimize the measurement error $\|\varepsilon\|$ while assuming a constant magnitude for the measured magnetic field. Thus, given N measurements, the problem of calibrating a magnetometer can be posed as the following optimization problem

$$\begin{aligned} & \underset{\substack{T_m, h_m \\ m_k, k=1, 2, \dots, N}}{\text{minimize}} && \sum_{k=1}^N \|y_{m_k} - T_m m_k - h_m\|^2 \\ & \text{subject to} && \|m_k\| = 1, \quad k = 1, 2, \dots, N \end{aligned} \quad (2.7)$$

All norms in this paper are two-norms unless it is indicated otherwise. In (2.7), without loss of generality, we assume the magnitude of the magnetic field is one. A penalty function corresponding to (2.7) is

$$J = \sum_{k=1}^N \left[\|y_{m_k} - T_m m_k - h_m\|^2 + \lambda (\|m_k\|^2 - 1)^2 \right] \quad (2.8)$$

where λ is a positive constant. It should be selected to balance the contribution of the two summands¹.

Minimizing (2.8) using gradient descent or Newton-Raphson methods require a good initial estimate of the unknowns T_m and h_m , otherwise they are very slow in convergence, if they converge at all. Finding an initial estimate is not trivial due to the uncertainty of soft-iron and hard-iron distortions; the authors in [38] and [12] propose a linear least-squares problem in order to find one. In [5] a solution to (2.7) by means of iterations of a least-square problem is proposed which excels in computational efficiency and convergence.

Similarly to [5], we propose an iterative solution to (2.7) based on the solution of a linear least-squares problem. We start with rewriting (2.4) in matrix form for all measurements

$$Y = LG + E \quad (2.9)$$

where

$$Y = \begin{bmatrix} y_{m_1} & y_{m_2} & \dots & y_{m_N} \end{bmatrix}, \quad L = \begin{bmatrix} T_m & h_m \end{bmatrix}$$

$$G = \begin{bmatrix} m_1 & m_2 & \dots & m_N \\ 1 & 1 & \dots & 1 \end{bmatrix} \quad \text{and} \quad E = \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & \dots & \varepsilon_N \end{bmatrix}$$

¹Typically it is selected to be in the order of $\|T_m\|$.

The system (2.9) has $3 \times N$ equations. Assuming an initial estimate of a full rank matrix G , every iteration of the algorithm begins with deriving T_m and h_m minimizing the total squared error $\|E^T E\|_F^2$. From least-squares method [32] we have

$$L = YG^T(GG^T)^{-1} \quad (2.10)$$

Using the updated values of T_m and h_m and (2.4) we define

$$\tilde{m}_k = T_m^{-1}(y_{m_k} - h_m), k = 1, 2, \dots, N \quad (2.11)$$

where we assume that T_m is invertible. This is a rational assumption as a non-invertible T_m would imply that not all three axes are expressed in the output of the sensor.

Since the magnitude of the magnetic field is independent of the measurement, and set to one for convenience, we update m_k as

$$m_k = \frac{\tilde{m}_k}{\|\tilde{m}_k\|} \quad (2.12)$$

As a metric of convergence we use the value of the penalty function J in (2.8). The magnetometer calibration algorithm is summarized in Algorithm 1.

Algorithm 1: Magnetometer Calibration Algorithm

- Step 1: Initialize $m_k = \frac{y_{m_k}}{\|y_{m_k}\|}, k = 1, 2, \dots, N$
and form matrix G
- Step 2: Solve for L using least-squares:

$$L = YG^T(GG^T)^{-1}$$
- Step 3: Extract T_m and h_m from L
- Step 4: $\tilde{m}_k = T_m^{-1}(y_{m_k} - h_m), k = 1, 2, \dots, N$
- Step 5: Update G using $m_k = \frac{\tilde{m}_k}{\|\tilde{m}_k\|}, k = 1, 2, \dots, N$
- Step 6: Calculate J
- Step 7: Repeat steps 2-6 until J is sufficiently small
-

2.4 Accelerometer's Calibration

The 3-axis accelerometer's calibration algorithm exploits the fact that measured magnitude of the specific force should be constant when the sensor is still, independently of the sensor's orientation. A popular calibration approach ([19], [2] and others) uses the fact that the measured magnitude of the specific force of a still 3-axis accelerometer should be constant. Assuming N measurements and using (2.2) the calibration problem is equivalent to minimizing (2.13); this is typically done by employing the gradient descent method.

$$\begin{aligned} & \underset{\substack{T_a, h_a \\ f_k, k=1, 2, \dots, N}}{\text{minimize}} && \sum_{k=1}^N \|y_{ak} - T_a f_k - h_a\|^2 \\ & \text{subject to} && \|f_k\| = 1, \quad k = 1, 2, \dots, N \end{aligned} \quad (2.13)$$

A penalty function corresponding to (2.13) is

$$J = \sum_{k=1}^N \left\{ \|y_{ak} - T_a f_k - h_a\|^2 + \lambda (\|f_k\|^2 - 1)^2 \right\} \quad (2.14)$$

where λ is a positive constant. It should be selected to balance the contribution of the two summands ².

In contrast to the magnetometer case, for the accelerometer's calibration we can find an initial estimate of the unknowns f , T_a and h_a . Under the reasonable assumption of small scale-factor and cross-coupling errors, an initial estimate of T_a is the 3×3 identity matrix ³. In a similar way, bias vector h_a is initialized to the 3×1 zero vector ($0_{3 \times 1}$).

Using the aforementioned initial estimate, the gradient descent method can minimize (2.14). However, as seen, optimization problem (2.13) share the same form with (2.7), the optimization problem derived for magnetometer calibration. Thus, magnetometer's calibration algorithm can also be used for accelerometer calibration as shown in Algorithm 2. Algorithm 2 is typically significantly faster in convergence and better in computational efficiency than the gradient descent one and this why it is preferred in our proposed calibration methodology here.

²Typically λ is selected to be in the order of $\|T_a\|$

³This relates to the fact that we use $\|f_k\| = 1$ and the assumption that the accelerometer's gain has been roughly pre-adjusted so that the specific force results in almost unit magnitude output.

Algorithm 2: Accelerometer Calibration

Step 1: Initialize $f_k = \frac{y_{a_k}}{\|y_{a_k}\|}$, $k = 1, 2, \dots, N$
 and form matrix G

Step 2: Solve for L using least-squares:

$$L = YG^T(GG^T)^{-1}$$

Step 3: Extract T_a and h_a from L

Step 4: $\tilde{f}_k = T_a^{-1}(y_{a_k} - h_a)$, $k = 1, 2, \dots, N$

Step 5: Update G using $f_k = \frac{\tilde{f}_k}{\|\tilde{f}_k\|}$, $k = 1, 2, \dots, N$

Step 6: Calculate J

Step 7: Repeat steps 2-6 until J is sufficiently small

2.5 Performance Evaluation of Accelerometer's and Magnetometer's Calibration Algorithm

To evaluate the performance of the proposed algorithm we used several different datasets recorded following the calibration procedure presented in [25]. In this Section we first evaluate the convergence of the proposed algorithm using five different datasets. Next we evaluate the quality of the resulting calibration parameters and the repeatability of the algorithm's results. Finally, we demonstrate the effect of the proposed data on the sensor's measurements.

2.5.1 Algorithm's Convergence

The convergence of five different datasets of accelerometer's data, recorded following the calibration procedure presented in [25] is shown in Figure 2.1. As seen in Figure 2.1, the cost function appears to be monotonic and requires only a few iterations of the algorithm to converge.

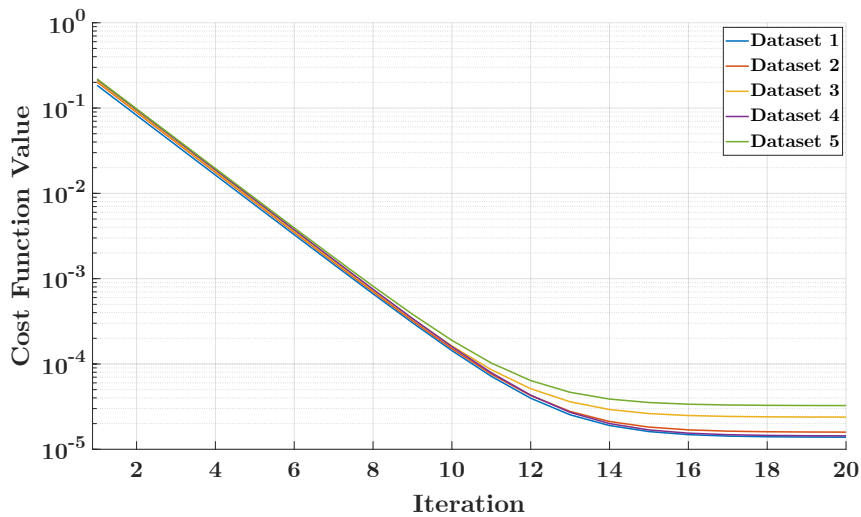


Figure 2.1: Convergence of accelerometer calibration algorithm

In Figure 2.2 the convergence of the proposed algorithm when using magnetometer's measurements is presented. Although the required iterations and the shape of the cost function are very different for each dataset, the algorithm converges monotonically for all five dataset.

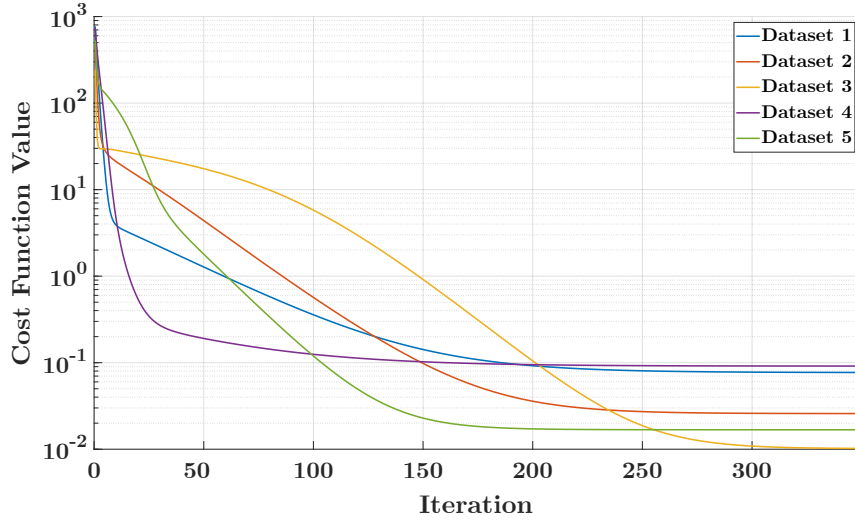


Figure 2.2: Convergence of magnetometer calibration algorithm

2.5.2 Measuring Distance Between Calibration Parameter Sets

Applying the calibration methodology to all five datasets, we expect some consistency between the calibration parameters derived from each dataset. The calibration parameter sets of the magnetometer and the accelerometer are pairs of a calibration matrix and an offset vector, (T_m, h_m) and (T_a, h_a) as shown in (2.4) and (2.2) respectively.

The offset vectors are defined uniquely in the proposed algorithms, in the sense that they are independent of the true values of the magnetic field, the specific force and the angular velocity respectively. Therefore, the distance between offset vectors derived using different datasets can be defined as the norm of their algebraic difference, i.e. $d(x_i, x_j) = \|x_i - x_j\|$.

The normalized distance \bar{d} is defined as the ratio of the average distance divided by the average norm of the vectors, i.e. for N datasets ($N = 5$ here) there are $\binom{N}{2}$ pairs and \bar{d} is given by

$$\bar{d}(x_1, x_2, \dots, x_N) = \frac{\left(\sum_{1 \leq i < j \leq N} d(x_i, x_j) \right) / \binom{N}{2}}{\left(\sum_{1 \leq i \leq N} \|x_i\| \right) / N}.$$

For the vectors sets h_m and h_a we calculated the distance \bar{d} based on five datasets. It is

$$\begin{aligned} \bar{d}(h_{m1}, h_{m2}, \dots, h_{m5}) &= 0.0208 \\ \bar{d}(h_{a1}, h_{a2}, \dots, h_{a5}) &= 0.0393 \end{aligned} \tag{2.15}$$

Defining the distance between calibration matrices derived using different datasets is more

tricky because the proposed algorithms consider the true values of the magnetic field and the specific force to be unknowns. They are derived along with the calibration matrices to minimize the random errors in (2.4) and (2.2) respectively.

Observe for example in (2.4) that if we replace T_m with $T_m Q$ and m with $Q^T m$, where Q is an orthogonal 3×3 matrix, i.e. $Q \in O(3)$, the resulting measurement y_m is unaltered. The same is true for (2.2).

Therefore the calibration matrices are derived subject to orthogonal⁴ multiplication uncertainty. To this end we define the distance between two calibration matrices (of the accelerometer, the magnetometer or the gyroscope) derived from different datasets as follows.

The distance between two 3×3 matrices T_1 and T_2 can be defined as:

$$D(T_1, T_2) = \min_{Q \in O(3)} \|QT_1 - T_2\|_F \tag{2.16}$$

The minimizing matrix Q can be calculated using the orthogonal Procrustes Theorem [27].

Similarly to the offset vectors, the normalized distance \bar{D} between a number N of calibration matrices is defined as

$$\bar{D}(X_1, X_2, \dots, X_N) = \frac{\left(\sum_{1 \leq i < j \leq N} D(X_i, X_j) \right) / \binom{N}{2}}{\left(\sum_{1 \leq i \leq N} \|X_i\|_F \right) / N}. \tag{2.17}$$

For the calibration matrix sets T_m and T_a we calculated the distance \bar{D} based on five datasets. It is

$$\begin{aligned} \bar{D}(T_{m1}, T_{m2}, \dots, T_{m5}) &= 0.0287 \\ \bar{D}(T_{a1}, T_{a2}, \dots, T_{a5}) &= 0.0018 \end{aligned} \tag{2.18}$$

Functions $d(\cdot, \cdot)$ and $D(\cdot, \cdot)$ represent the distance among the offset vectors and the distance among the calibration matrices derived from different datasets respectively. Thus, the small values of the normalized average distances (2.15) and (2.18) indicate good repeatability of the proposed calibration algorithm i.e. the algorithm return similar calibration parameters for different dataset inputs.

⁴Note that the uncertainty is not extended to a larger matrix set as it must preserve the norm of all possible true values.

2.5.3 The Effect of Calibration on Sensors' Measurements

For both accelerometer and magnetometer, the effectiveness of the calibration algorithm can be deduced from the true values of the specific force \tilde{f} and the magnetic field \tilde{m} having unit magnitude. In Figures 2.3 and 2.4 the normalized magnitude of both raw and calibrated accelerometer and magnetometer measurements are presented.

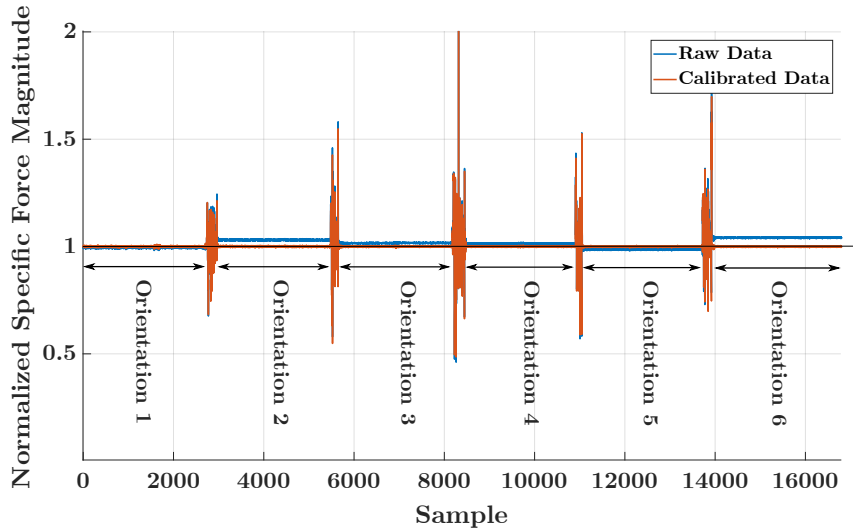


Figure 2.3: Normalized magnitude of raw and calibrated accelerometer measurements in six still orientations.

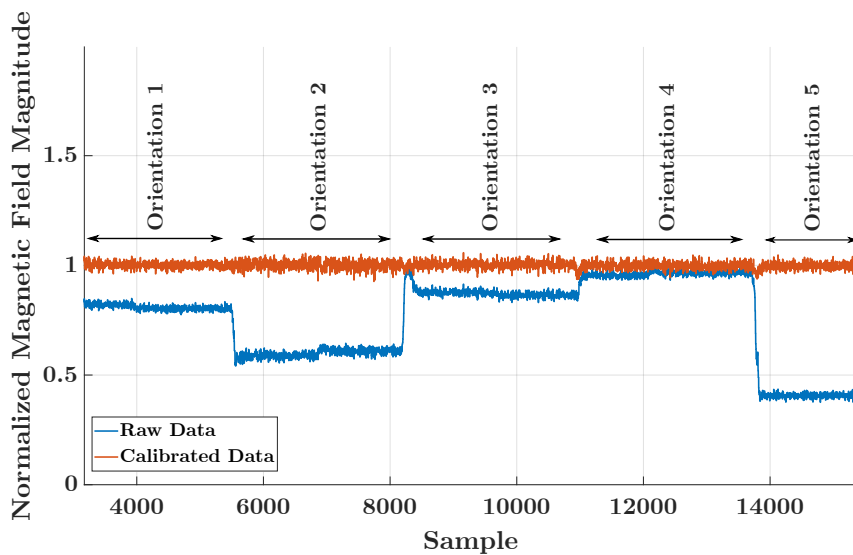


Figure 2.4: Normalized magnitude of raw and calibrated magnetometer measurements in five still orientations.

2.6 Accelerometer's and Magnetometer's Axes Alignment

Inertial and magnetic sensors are often used in combination in many applications including, navigation [6] and attitude estimation [20], healthcare systems [28], gaming and entertainment devices [4], space exploration and many other industrial and commercial ones. In such cases it is crucial for their sensitivity axes (coordinate frames) to be aligned.

Assuming that the coordinate frames of the accelerometer and the magnetometer are $\{x_a, y_a, z_a\}$ and $\{x_m, y_m, z_m\}$, respectively, aligning the two coordinate frames comes down to deriving a rotation matrix $R_M^A \in SO(3)$ such that $R_M^A \{x_m, y_m, z_m\} = \{x_a, y_a, z_a\}$ as shown in Figure 2.5.

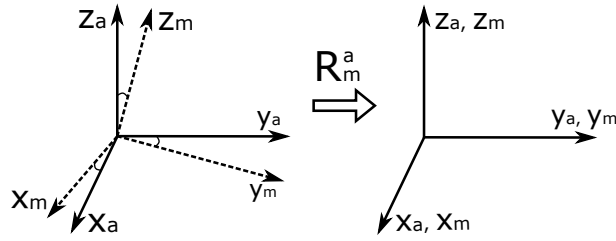


Figure 2.5: Axes alignment.

Axes alignment algorithms require an accurately known magnetic field to be used as reference in order to derive R_M^A . When cost is of no concern, the reference magnetic field is generated using expensive laboratory equipment. For low-cost sensors however this is impractical due to incommensurate extra cost.

Several works propose axes alignment methods which require no special piece of equipment [39, 12, 15, 25, 21, 8, 17, 24]. A standard approach is to exploit the magnetic inclination phenomenon as reference in order to align the axes of an accelerometer and a magnetometer. Magnetic inclination (or magnetic dip) is the angle between the horizon and the Earth's magnetic field lines as shown in Figure 2.6.

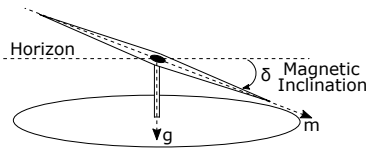


Figure 2.6: Magnetic inclination.

It varies with location and time and the sine of it is the inner product of the normalized gravity and the magnetic field vectors.

$$s_\delta \triangleq \sin(\delta) = \frac{g^T m}{\|g\| \|m\|}. \quad (2.19)$$

Existing axes alignment algorithms, e.g., [25, 24, 12, 8], use (2.19) and accelerometer's and magnetometer's measurements to form an optimization problem for deriving R_M^A . To do so, some

of them [25, 24] form a cost-plus-penalty function associated with (2.19) which is then minimized using gradient descent or Newton-Raphson method.

In this work, we propose a complete analytical solution to the axes alignment problem guaranteeing the orthogonality of the axes alignment rotation matrix, with best-of-class-accuracy. This is done by introducing a new formulation of the axes alignment problem which transforms the original constrained optimization problem to a smooth unconstrained one.

The proposed method derives both the magnetic inclination angle and the axes alignment rotation matrix in closed-form. Due to its analytical nature, the proposed method gives highly accurate results, comparable to the best ones achieved by existing iterative methods, requiring however significantly lower computational resources.

2.6.1 Problem Statement

Consider a 3-axis accelerometer and a 3-axis magnetometer, both fixed on the same rigid platform, and denote their coordinate frames as $\{A\}$ and $\{M\}$ respectively ⁵.

Now suppose that accelerometer's measurement g_k^A and magnetometer's measurement m_k^M are taken simultaneously, when the rigid platform is still (only gravitational force), for $k = 1, 2, \dots, K$ with $K \geq 9$. Platform's orientation changes with k and every measured vector is expressed in the coordinate frame of the corresponding sensor.

Let the gravity vector g and magnetic field m be expressed in a fixed inertial coordinate frame $\{I\}$. For every $k = 1, 2, \dots, K$ there is a rotation matrix $Q_k \in SO(3)$ transforming vectors from the $\{I\}$ -frame to the $\{A\}$ -frame. Then the gravity and magnetic field vectors expressed in the $\{A\}$ -frame are written as

$$g_k^A = Q_k g \quad \text{and} \quad m_k^A = Q_k m \quad (2.20)$$

for $k = 1, 2, \dots, K$, respectively. Solving (2.20) for g and m and replacing them in (2.19),

$$\sin(\delta) = \frac{(g_k^A)^T Q_k Q_k^T m_k^A}{\|Q_k^T g_k^A\| \|Q_k^T m_k^A\|} = \frac{(g_k^A)^T m_k^A}{\|g_k^A\| \|m_k^A\|} \quad (2.21)$$

for $k = 1, 2, \dots, K$, where we exploited the orthogonality of Q_k and the rotational invariance of the Euclidean norm.

In (2.21), the accelerometer's measurement g_k^A is known. The measured magnetic field m_k^M however is naturally expressed in the $\{M\}$ -frame and so we need to transform it to the $\{A\}$ -frame. To this end, let $R_M^A \in SO(3)$ be the transformation matrix from $\{M\}$ -frame to $\{A\}$ -frame, then

$$m_k^A = R_M^A m_k^M, \quad k = 1, 2, \dots, K \quad (2.22)$$

Combining (2.21) and (2.22) and using again the rotational invariance of the Euclidean norm,

⁵Both coordinate frames are considered to be right-handed.

we get that

$$\sin(\delta) = \frac{(g_k^A)^T R_M^A m_k^M}{\|g_k^A\| \|m_k^M\|}, \quad k = 1, 2, \dots, K \quad (2.23)$$

In (2.23), g_k^A and m_k^M are the known accelerometer's and magnetometer's measurements, respectively, while both R_M^A and inclination angle, δ , are unknown. A standard approach to derive them is to form and solve the optimization problem

$$\begin{aligned} & \underset{R_M^A, \delta}{\text{minimize}} && \sum_{k=1}^K \left(\sin(\delta) - \frac{(g_k^A)^T R_M^A m_k^M}{\|g_k^A\| \|m_k^M\|} \right)^2 \\ & \text{subject to} && R_M^A \in SO(3) \\ & && \delta \in \left[-\frac{\pi}{2}, \frac{\pi}{2} \right] \end{aligned} \quad (2.24)$$

2.6.2 Prior Art and its Limitations

A typical approach to solve (2.24) is to minimize an associated cost-plus-penalty function using the gradient descent or the Newton-Raphson method. In [25, 24] the authors use the following cost-plus-penalty function, J_{CP} , associated with (2.24) and incorporating a weighted penalty term capturing the non-orthogonality of R ⁶

$$J_{CP}(R, s_\delta) = \sum_{k=1}^K \left(s_\delta - g_k^T R m_k \right)^2 + \lambda \|RR^T - I\|_F^2. \quad (2.25)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. In using (2.25) in [25, 24], special care should be given to the selection of the weighting parameter, λ , in order to ensure both the approximate orthogonality of R and the (fast) converge of the minimization method.

Following the iterative optimization approaches [25, 24] and (2.25), we first consider the case when both R and s_δ are initialized without any prior knowledge, as the identity matrix and zero respectively. Using the Newton-Raphson method and a set of sensors' measurements, we minimize (2.25) for multiple values of λ . To assess the distance from orthogonality of the derived matrices R , we first define the nearest orthogonal matrix to R as [27]

$$R_O = UV^T \quad (2.26)$$

where $U \in O(3)$ and $V \in O(3)$ are defined via a singular value decomposition (SVD) of $R = U\Sigma V^T$. Then the distance of R from orthogonality is defined as

$$D_O(R) \triangleq \|R - R_O\| = \|R - UV^T\|. \quad (2.27)$$

⁶Note that when the initial condition of the Newton-Raphson is far from the final solution, an extra term in (2.25) is required to force the determinant of R to be equal to one and thus $R \in SO(3)$. However, if Newton-Raphson initial condition is near to the final solution, this term may be omitted.

The convergence of J_{CP} using the Newton-Raphson for different values of λ is shown in Figure 2.7.

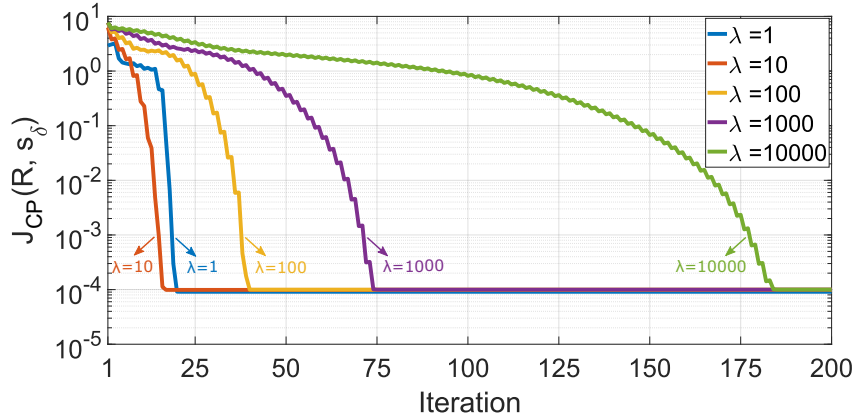


Figure 2.7: Convergence of Newton-Raphson for different values of the weighting factor, λ (Fastest convergence corresponds to $\lambda = 10$).

The numbers of iterations for Newton-Raphson to converge (J_{CP} to drop below 10^{-4}) and the distance of the derived matrix R from orthogonality are presented in Figure 2.8 as functions of λ .

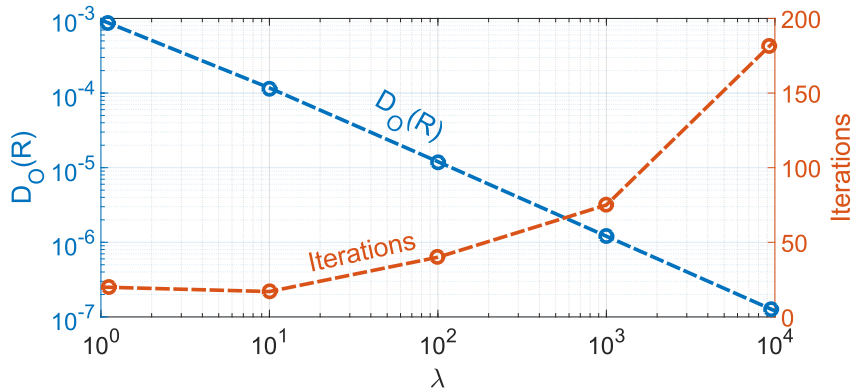


Figure 2.8: Distance of R from orthogonality $D_O(R)$ when Newton-Raphson has converged, and, number of iterations required for convergence, as functions of weighting factor λ .

We observe that larger values of λ result in R closer to orthogonality. However, Newton-Raphson requires more iterations to converge for larger λ , implying a trade-off between the orthogonality of R and computational efficiency.

A better trade-off is obtained when the results of the single-step method in [30] are used to initialize the Newton-Raphson method. As seen in Figures 2.9 and 2.10, Newton-Raphson converges after only two iterations even when large values of λ are used. However, in this case, the computational complexity of the single-step method of [30] must be also taken into account.

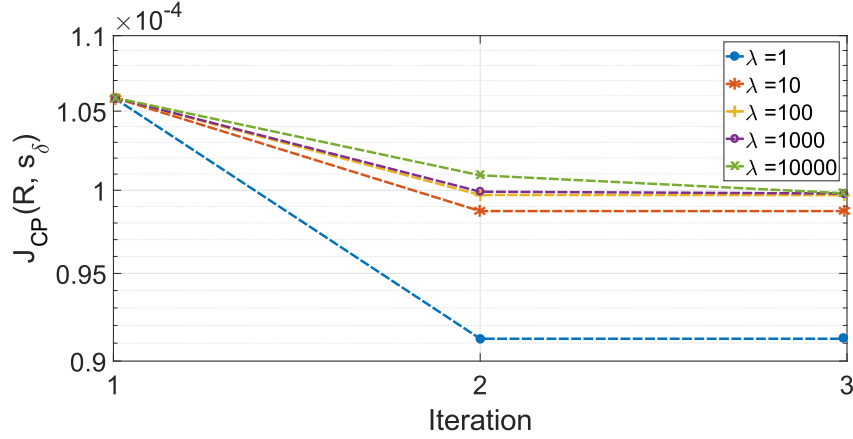


Figure 2.9: Convergence of Newton-Raphson for different values of the weighting factor, λ , when the single-step method in [30] is used for initialization.

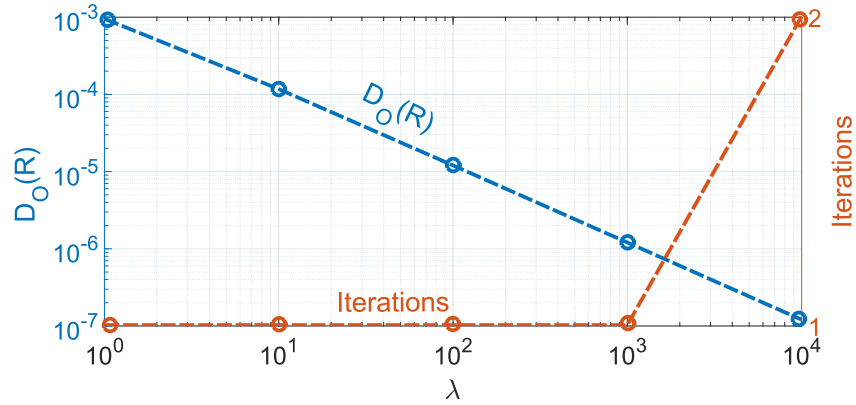


Figure 2.10: Distance of R from orthogonality and Newton-Raphson iterations until convergence for different values of the weighting factor, λ , when the single-step method in [30] is used for initialization.

2.6.3 The Proposed Algorithm

The proposed method converts the constrained optimization problem (2.24) to an unconstrained one which is solved using analytical iterations of the Newton-Raphson method. Furthermore, using a good initial estimate of the point of minimum, as done later in this section, implies that only one iteration is sufficient to achieve a very accurate result.

To convert the constrained problem (2.24) into an unconstrained one, we first derive the optimal value of s_δ analytically and formulate an equivalent optimization problem with the single unknown R . To do so, consider the cost function of (2.24)

$$J(R, s_\delta) = \sum_{k=1}^K (s_\delta - g_k^T R m_k)^2 \quad (2.28)$$

and note that it is quadratic with respect to s_δ . Defining the 9×1 vector $V_R = \text{vec}(R)$ and using the identity $g_k^T R m_k = (m_k \otimes g_k)^T \text{vec}(R)$ we write

$$J(R, s_\delta) = K s_\delta^2 - 2 s_\delta \underline{1}^T A V_R + V_R^T A^T A V_R \quad (2.29)$$

where \otimes is the Kronecker's product [11], $\underline{1}$ is the $K \times 1$ vector of ones and the $K \times 9$ matrix A is

$$A = \begin{bmatrix} (m_1 \otimes g_1)^T \\ (m_2 \otimes g_2)^T \\ \vdots \\ (m_K \otimes g_K)^T \end{bmatrix}. \quad (2.30)$$

We define the minimum of $J(s_\delta, R)$ with respect to s_δ , i.e.,

$$J_1(R) \triangleq \min_{|s_\delta| \leq 1} J(R, s_\delta) \quad (2.31)$$

and observe that the unconstrained point of minimum is

$$s_\delta^* = \frac{1}{K} \underline{1}^T A V_R. \quad (2.32)$$

Note that (2.32) can also be written as

$$s_\delta^* = \frac{1}{K} \sum_{i=1}^K (g_i^T R m_i). \quad (2.33)$$

Following our assumption that $\|g_i\| = \|m_i\| = 1$ for all $i = 1, 2, \dots, K$ and the fact that $\|\cdot\|_2$ norm is rotational invariant, by applying the Cauchy–Schwarz inequality to (2.33) we get $|s_\delta^*| \leq 1$ and so s_δ^* is feasible and the global minimum of (2.31).

Replacing (2.32) into (2.29), $J_1(R)$ is conveniently written as

$$J_1(R) = \frac{1}{2} V_R^T B V_R \quad (2.34)$$

where $B = 2 \left(A^T A - \frac{1}{K} A^T \underline{1} \underline{1}^T A \right)$ is a 9×9 symmetric matrix. Note that by the definition of J_1 we have

$$\min_{R \in SO(3), |s_\delta| \leq 1} J(R, s_\delta) = \min_{R \in SO(3)} J_1(R) \quad (2.35)$$

where the minimum exists since the cost function J_1 is continuous and $SO(3)$ is compact.

Let $R_* \in SO(3)$ be a point of global minimum of J_1 i.e.

$$J_1(R_*) = \min_{R \in SO(3)} J_1(R) \quad (2.36)$$

and let $R_0 \in SO(3)$ be an initial estimate of R_* . An improved estimate can always be expressed

as $R = PR_0$, for some $P \in SO(3)$. Moreover we can write P as a sequence of three Euler rotations, i.e.,

$$P = P(x) \triangleq R_z(\phi)R_y(\psi)R_x(\theta) \quad (2.37)$$

where ϕ , ψ and θ are the yaw, pitch and roll rotation angles respectively, $x \triangleq [\phi, \psi, \theta]^T$ and

$$\begin{aligned} R_z(\phi) &= \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ R_y(\psi) &= \begin{bmatrix} \cos(\psi) & 0 & -\sin(\psi) \\ 0 & 1 & 0 \\ \sin(\psi) & 0 & \cos(\psi) \end{bmatrix} \\ R_x(\theta) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \end{aligned} \quad (2.38)$$

The function $P : [0, 2\pi]^3 \rightarrow SO(3)$ is surjective and so $R = P(x)R_0$ can take any matrix value in $SO(3)$, [10].

The above convert the original optimization problem to the one of deriving x such that $P(x)R_0 = R_*$. To proceed further, it is convenient to define the cost as a function of x , i.e.,

$$J_2(x) \triangleq J_1(P(x)R_0) = \frac{1}{2}V_R(x)^T B V_R(x). \quad (2.39)$$

where $V_R(x) = \text{vec}(R(x)) = \text{vec}(P(x)R_0)$.

Assume that R_0 and $R = PR_0$ are close to R_* i.e. $\|R_0 - R_*\|_F$ and $\|R - R_*\|_F$ are small⁷. Then P is close to the identity matrix and so there exists a small x such that $P = P(x)$, [10]. This along with the smoothness of the functions involved motivates the use of minimization methods based on Taylor expansion like Newton-Raphson.

To minimize $J_2(x)$ we have to derive x such that $\partial J_2 / \partial x = 0$. To do so, we start from $x = 0$ implying $P(x) = I$ and cost $J_2(0)$, apply one iteration (or more) of Newton-Raphson method, and derive the new value of x as

$$x = - \left(\frac{\partial^2 J_2}{\partial x \partial x^T} \Big|_{x=0} \right)^{-1} \frac{\partial J_2}{\partial x} \Big|_{x=0}. \quad (2.40)$$

The cost gradient is

$$\frac{\partial J_2}{\partial x} = \left[\frac{\partial J_2}{\partial \phi}, \frac{\partial J_2}{\partial \psi}, \frac{\partial J_2}{\partial \theta} \right]^T \quad (2.41)$$

⁷With respect to the Frobenius or any other rotational invariant matrix norm.

and the Hessian matrix is symmetric and written as

$$\frac{\partial^2 J_2}{\partial x \partial x^T} = \begin{bmatrix} \frac{\partial^2 J_2}{\partial \phi^2} & \frac{\partial^2 J_2}{\partial \phi \partial \psi} & \frac{\partial^2 J_2}{\partial \phi \partial \theta} \\ \frac{\partial^2 J_2}{\partial \phi \partial \psi} & \frac{\partial^2 J_2}{\partial \psi^2} & \frac{\partial^2 J_2}{\partial \psi \partial \theta} \\ \frac{\partial^2 J_2}{\partial \phi \partial \theta} & \frac{\partial^2 J_2}{\partial \psi \partial \theta} & \frac{\partial^2 J_2}{\partial \theta^2} \end{bmatrix} \quad (2.42)$$

because of the continuity of all second derivatives.

We derive the first and second derivatives at $x = 0$ analytically recalling that $B^T = B$. From (2.39) and for $s, q \in \{\phi, \psi, \theta\}$ we have that

$$\frac{\partial J_2}{\partial q} = V_R^T B \frac{\partial V_R}{\partial q} \quad (2.43)$$

and

$$\frac{\partial^2 J_2}{\partial s \partial q} = \frac{\partial V_R^T}{\partial s} B \frac{\partial V_R}{\partial q} + V_R^T B \frac{\partial^2 V_R}{\partial s \partial q} \quad (2.44)$$

From the definition $V_R = \text{vec}(R)$ we have that

$$\frac{\partial V_R}{\partial q} = \text{vec} \left(\frac{\partial R}{\partial q} \right) \quad (2.45)$$

and

$$\frac{\partial^2 V_R}{\partial s \partial q} = \text{vec} \left(\frac{\partial^2 R}{\partial s \partial q} \right). \quad (2.46)$$

Moreover, since $R(x) = P(x)R_0 = R_z(\phi)R_y(\psi)R_x(\theta)R_0$, from (2.38) it is

$$\left. \frac{\partial R}{\partial q} \right|_{x=0} = P_q R_0 \quad (2.47)$$

for $q \in \{\phi, \psi, \theta\}$, and,

$$\left. \frac{\partial^2 R}{\partial s \partial q} \right|_{x=0} = P_s P_q R_0 \quad (2.48)$$

for the ordered pairs

$$(s, q) \in \{(\phi, \phi), (\phi, \psi), (\phi, \theta), (\psi, \psi), (\psi, \theta), (\theta, \theta)\}, \quad (2.49)$$

where

$$\begin{aligned} P_\phi &= \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ P_\psi &= \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}. \\ P_\theta &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \end{aligned} \quad (2.50)$$

Combining (2.45) with (2.47) and (2.46) with (2.48), respectively gives

$$\left. \frac{\partial V_R}{\partial q} \right|_{x=0} = (I_3 \otimes P_q) V_{R_0} \quad (2.51)$$

and

$$\left. \frac{\partial^2 V_R}{\partial s \partial q} \right|_{x=0} = (I_3 \otimes (P_s P_q)) V_{R_0} \quad (2.52)$$

where $V_{R_0} = \text{vec}(R_0)$. Finally, replacing (2.51) into (2.43) gives

$$\left. \frac{\partial J_2}{\partial q} \right|_{x=0} = V_{R_0}^T B (I_3 \otimes P_q) V_{R_0}. \quad (2.53)$$

Similarly, replacing (2.51) and (2.52) into (2.44) gives

$$\begin{aligned} \left. \frac{\partial^2 J_2}{\partial s \partial q} \right|_{x=0} &= V_{R_0}^T (I_3 \otimes P_s^T) B (I_3 \otimes P_q) V_{R_0} \\ &\quad + V_{R_0}^T B (I_3 \otimes (P_s P_q)) V_{R_0} \end{aligned} \quad (2.54)$$

Note that (2.54) is valid (only) for the six (s, q) pairs in (2.49).

2.6.4 Finding an Initial Condition

The proper selection of the initial matrix R_0 is crucial for achieving (fast) convergence. To this purpose we recommend using as R_0 the approximate closed-form solution of (2.24) derived in [30]. This is done as follows.

Using identity $\text{vec}(AXB) = (B^T \otimes A)\text{vec}(X)$, [11], where \otimes is the Kronecker's product, and the unit magnitude assumption, from (2.23) we get

$$s_\delta = (m_k^M \otimes g_k^A)^T \text{vec}(R) \quad (2.55)$$

for $k = 1, 2, \dots, K$. Then we express (2.55) in matrix form as

$$s_\delta \underline{1} = A \text{vec}(R) \quad (2.56)$$

We assume that the measurements are such that A is of full rank, i.e. $\text{rank}(A) = 9$. This along with the fact that $\text{vec}(R) \neq 0$, since $R \in SO(3)$, and (2.56) imply that

$$s_\delta \neq 0. \quad (2.57)$$

We solve (2.56) (in the least squares sense [32])⁸ to derive

$$\text{vec}(R) = s_\delta (A^T A)^{-1} A^T \underline{1}, \quad (2.58)$$

where the equation is approximate when the measurements are real. Next, we split the 9×1 vector $(A^T A)^{-1} A^T \underline{1}$ into three 3×1 vectors h_1 , h_2 and h_3 as

$$(A^T A)^{-1} A^T \underline{1} = \begin{bmatrix} h_1^T & h_2^T & h_3^T \end{bmatrix}^T. \quad (2.59)$$

Defining matrix H as

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} \quad (2.60)$$

and using (2.58)-(2.60) we express matrix R as

$$R = s_\delta H. \quad (2.61)$$

Note that in (2.61) matrix H is known but s_δ is not. To calculate s_δ we use the fact that $R \in SO(3)$ and so $\det(R) = 1$ which combined with (2.61) gives $\det(H) \neq 0$ and

$$1 = s_\delta^3 \det(H) \quad (2.62)$$

and so

$$s_\delta = \frac{1}{\sqrt[3]{\det(H)}}, \quad (2.63)$$

where the cubic root is constrained in the real numbers.

Although (2.63) is a compact expression, it is not as accurate with real data as the one we derive from the fact that $\|R\|_F = \sqrt{3}$ for $R \in SO(3)$, [11]. Combining it with (2.61) implies

$$|s_\delta| = \frac{\sqrt{3}}{\|H\|_F} \quad (2.64)$$

⁸With real measurements (2.55) is approximate and (2.56) is considered as a least squares problem with solution (2.58) where R is expected to be close to but not necessarily in $SO(3)$.

and the sign of s_δ can be recovered from (2.62) as

$$\text{sgn}(s_\delta) = \text{sgn}(\det(H)). \quad (2.65)$$

Consider a Singular Value Decomposition (SVD) of matrix H , i.e. $H = U\Sigma V^T$, where $U, V \in O(3)$ and Σ is the diagonal matrix $\Sigma = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$, with $\sigma_1 \geq \sigma_2 \geq \sigma_3 > 0$ since $\det(H) \neq 0$. It is $\|H\|_F = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}$, [11], which combined with (2.64) and (2.65) gives

$$s_\delta = \text{sgn}(\det(H)) \sqrt{\frac{3}{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}}. \quad (2.66)$$

By substituting (2.66) into (2.61) we get R , which ideally belongs to $SO(3)$. When using real measurements however, R may be close to but not necessarily in $SO(3)$. Thus, instead of R we derive and use the nearest special orthogonal matrix $\hat{R} \in SO(3)$ to $R = s_\delta H$ defined as the solution of the optimization problem

$$\hat{R} \triangleq \underset{Q \in SO(3)}{\text{argmin}} \|R - Q\|_F. \quad (2.67)$$

Even with real measurements, we expect that $\det(R) > 0$ which guarantees the uniqueness of the solution of the Orthogonal Procrustes Problem [27] in $O(3)$,

$$\bar{R} = \underset{Q \in O(3)}{\text{argmin}} \|R - Q\|_F. \quad (2.68)$$

The solution of (2.68) is $\bar{R} = \text{sgn}(s_\delta)UV^T$, [27]. It can be derived using the previously calculated SVD of H leading to an SVD of $R = (\text{sgn}(s_\delta)U)(|s_\delta|\Sigma)V^T$ via (2.61).

Also note that $\text{sgn}(\det(H)) = \text{sgn}(\det(UV^T))$ which along with (2.61) and (2.66) imply that $\det(\bar{R}) = 1$ and therefore $\hat{R} = \bar{R} \in SO(3)$ is the unique solution of (2.67), i.e.

$$\hat{R} = \text{sgn}(\det(H))UV^T \quad (2.69)$$

The complete proposed method using only one analytical iteration of the Newton-Raphson (which is typically sufficient) is summarized in Algorithm 3.

Algorithm 3: Proposed Method

-
- 1: Use normalized g_k and m_k to form matrix A in (2.30)
 - 2: Verify that A is of full rank
 - 3: Use (2.59), (2.60) and (2.69) to calculate R_0 as in [30]
 - 4: Calculate the gradient vector using (2.41) and (2.53)
 - 5: Calculate the Hessian matrix using (2.42) and (2.54)
 - 6: Calculate x from (2.40)
 - 7: Use x to calculate $P(x)$ using (2.37) and (2.38)
 - 8: Calculate $R = P(x)R_0 \in SO(3)$.
-

2.6.5 Evaluation of the Proposed Method

Let $R_M^A \in SO(3)$ be the frame transformation matrix rotating the magnetometer's coordinate frame into the accelerometer's one. To evaluate the accuracy and computational efficiency of the proposed method we have to compare the derived axes alignment matrix, R , to the actual one, R_M^A , which we assume to know accurately in advance.

However, the accuracy with which one can measure R_M^A using laboratory equipment is orders of magnitude worse than the expected accuracy of the proposed method. Therefore, we artificially generated 1000 datasets with pre-selected $R_M^A \in SO(3)$, according to the calibration procedure introduced in [25]. This included the random errors (noise) of the sensors and the associated instrumentation, according to typical characteristics of commercial devices.

To generate the 1000 datasets, we first randomly generated 1000 values of $R_A^M = (R_M^A)^T \in SO(3)$. For every one of them we followed the steps: 1) First we generated two random unit vectors, g_1^A and m_1^A representing the gravity and the magnetic field in the accelerometer's $\{A\}$ -frame. 2) We rotated both vectors 11 times according to [25] to generate $\{g_i^A\}_{i=2}^{12}$ and $\{m_i^A\}_{i=2}^{12}$. 3) To express the magnetic field vectors $\{m_i^A\}_{i=1}^{12}$ in the magnetometer's $\{M\}$ -frame, we rotate them once more using R_A^M to get $\{m_i^M\}_{i=1}^{12}$. 4) Finally a sequence of band-limited white noise was added to the dataset following typical sensors' and measuring procedure's specifications.

We compare our method's accuracy and execution time to those of 1) A gradient descent based method using (2.25), 2) A Newton-Raphson based method using (2.25), 3) A Newton-Raphson based method using (2.25), initialized using the solution of the single-step method presented in [30] and 4) The single step method of [30] alone.

Each of the aforementioned methods was run for every one of the 1000 generated datasets. For the iterative methods, based on the gradient descent and the Newton Raphson, the parameter λ of the cost function (2.25) was set to $\lambda = 1000$, to ensure the orthogonality of the derived matrix R according to Figures 2.8 and 2.10. We compared the derived matrix R , of each method, with the true rotation matrix $R_M^A = (R_A^M)^T$ used to generate the data. To quantify their difference, we used the error metric

$$\varepsilon = \|R - R_M^A\| \quad (2.70)$$

In the ideal case of perfect axes alignment, i.e., $R = R_M^A$, it is $\varepsilon = 0$. The mean value (μ_ε) and variance (σ_ε^2) of ε for every method is presented in Table 2.1.

Method	μ_ε	σ_ε^2	Time (ms)
GD	$7.16 \cdot 10^{-4}$	$9.90 \cdot 10^{-8}$	63.15
NR	$7.17 \cdot 10^{-4}$	$9.95 \cdot 10^{-8}$	44.53
NR & [30]	$7.15 \cdot 10^{-4}$	$9.92 \cdot 10^{-8}$	1.34
Single-Step [30]	$11.60 \cdot 10^{-4}$	$56.53 \cdot 10^{-8}$	0.25
Proposed	$7.14 \cdot 10^{-4}$	$9.94 \cdot 10^{-8}$	0.49

Table 2.1: Mean value and variance of the error ε of the proposed method, a gradient descent (GD) based method, a Newton-Raphson (NR) based method, a Newton-Raphson based method initialized using the solution of [30] and the single-step method of [30] alone.

As seen in Table 2.1, the gradient descent and the Newton-Raphson based methods alone yield in accurate results requiring however significant computational effort. The single-step method of [30] has much better computational efficiency but it is a little less accurate. The proposed method excels in both accuracy and computational efficiency. It provides accurate results, similar to those of the computationally heavy, iterative optimization methods, while it requires significantly less computational resources.

While artificially generated data are appropriate to evaluate the accuracy and computational efficiency of the proposed algorithm, they do not incorporate the non-idealities expected in real world measurements. Although we included random noise in the artificially generated data, other errors, such as residual calibration errors (of the sensors individually) could degrade the proposed algorithm's performance.

To demonstrate the resilience of the proposed algorithm to such effects, we recorded five different datasets of accelerometer's and magnetometer's measurements. To this end, we used a measurement device based on the LSM9DS1 system-in-package by STMicroelectronics which includes both a three-axis accelerometer and a three-axis magnetometer. Some important performance characteristics of the two sensors and the developed measurement device are presented in Table 2.2.

Specification	Value
Measurement Range (A)	$\pm 16g$
Measurement Range (M)	$\pm 4Gauss$
Sampling Rate (A)	$238Hz$
Sampling Rate (M)	$80Hz$
Resolution (A, M)	$16Bits$

Table 2.2: Performance characteristics of the accelerometer (A) and the magnetometer (M) included in the designed measurement device.

All datasets were recorded away from magnetic disturbances (the constant earth's magnetic field was used as reference) following the calibration procedure introduced in [25]. Specifically, to

record each dataset we placed the measurement device by hand in twelve different orientations, as suggested in [25]. In each orientation we recorded several measurements while the sensor was still and used averaging to obtain twelve pairs of accelerometer's and magnetometer's measurements corresponding to the twelve orientations.

In this case of real sensors' data, the true matrix R_M^A is not known. Thus, in order to evaluate the accuracy of the proposed algorithm and compare it to the that of the existing ones, we use the cost-plus-penalty function of (2.25) as a metric of the residual error.

In Table 2.3 we used five different datasets (D1-D5) to compare our method's residual error to that of 1) A gradient descent based method using (2.25), 2) A Newton-Raphson based method using (2.25), 3) A Newton-Raphson based method using (2.25), initialized using the solution of the single-step method presented in [30] and 4) The single step method of [30].

Again, for the iterative methods, based on the gradient descent and the Newton Raphson, the parameter λ of the cost function (2.25) was set to $\lambda = 1000$, to ensure the orthogonality of the derived matrix R according to Figures 2.8 and 2.10.

Method	$J_{CP}(R, s_\delta) \cdot 10^4$				
	D1	D2	D3	D4	D5
GD	7.02	7.33	7.12	6.40	7.49
NR	7.02	7.35	7.24	6.38	7.37
NR & [30]	7.01	7.32	7.16	6.39	7.53
Single-Step [30]	9.36	9.21	9.12	8.42	9.68
Proposed	7.01	7.33	7.09	6.37	7.72

Table 2.3: Residual error of the proposed method, a gradiend descent (GD) based method, a Newton-Raphson (NR) based method, a Newton-Raphson based method initialized using the solution of [30] and the "single-step" method of [30] evaluated using five different datasets (D1-D5) of real sensors' data.

2.7 Gyroscope's Calibration

This Section introduces a new approach to gyroscope calibration and provides the associated algorithm. It applies to the case where a 3-axis gyroscope, a 3-axis accelerometer and a 3-axis magnetometer are fixed on the same, rigid platform.

Assuming that the accelerometer and the magnetometer have already been calibrated and aligned, we use them to calculate the rotation of the platform between two still positions. The proposed approach is based on the fact that this rotation should be identical to that derived from the gyroscope, when the last one is also calibrated. Note that using the joint accelerometer - magnetometer rotation as reference for the gyroscope's calibration, the algorithm also aligns the axes of the gyroscope with those of the other two sensors.

Therefore, the new approach is comprised of a) the derivation of the rotation from the accelerometer and magnetometer data, b) the parametric derivation of the rotation from the gyroscope data, and, c) the optimization algorithm which equates the two of them.

2.7.1 Rotation From Accelerometer and Magnetometer Data

Assume a rotation of the platform between two still positions. Let f_{begin} and f_{end} be the 3×1 accelerometer's measurement vectors before and after the rotation, while the platform is still. Similarly, let m_{begin} and m_{end} be the 3×1 magnetometer's measurement vectors accordingly.

Assuming that f_{begin} , f_{end} , m_{begin} and m_{end} have been derived using calibrated accelerometer and magnetometer according to the proposed algorithms in Sections 2.4 and 2.3, the angle between f_{begin} and m_{begin} is the same with the angle between f_{end} and m_{end} and all four vectors are of unit norm. The above allow us to use the TRIAD algorithm [3] [29] to find a rotation matrix $R \in SO(3)$ such that $Rf_{begin} = f_{end}$ and $Rm_{begin} = m_{end}$.

Given the unit vectors f_{begin} , f_{end} , m_{begin} and m_{end} , the TRIAD algorithm begins by constructing two triads of orthonormal column vectors according to

$$\begin{aligned} a_1 &= f_{begin}, & a_2 &= (f_{begin} \times m_{begin}) / \|f_{begin} \times m_{begin}\| \\ a_3 &= (f_{begin} \times (f_{begin} \times m_{begin})) / \|f_{begin} \times m_{begin}\| \end{aligned}$$

and

$$\begin{aligned} b_1 &= f_{end}, & b_2 &= (f_{end} \times m_{end}) / \|f_{end} \times m_{end}\| \\ b_3 &= (f_{end} \times (f_{end} \times m_{end})) / \|f_{end} \times m_{end}\| \end{aligned}$$

The matrix R is derived as

$$R = [b_1 \ b_2 \ b_3] [a_1 \ a_2 \ a_3]^T$$

It is convenient to consider the application of the TRIAD algorithm as a function R_{am} , i.e.

$$R = R_{am}(f_{begin}, f_{end}, m_{begin}, m_{end}) \quad (2.71)$$

2.7.2 Rotation From Gyroscope Data

Using K sequential gyroscope's measurements sampled at rate τ_s , we can calculate the rotation matrix representing the sensor's body frame rotation from time t to time $t + K\tau_s$. Let $\omega_k = [\omega_{xk} \ \omega_{yk} \ \omega_{zk}]^T$ be the k^{th} sample of the gyroscope's output. Using the entries of ω_k , we define the skew symmetric matrix function

$$\Omega(\omega_k) = \begin{bmatrix} 0 & -\omega_{zk} & \omega_{yk} \\ \omega_{zk} & 0 & -\omega_{xk} \\ -\omega_{yk} & \omega_{xk} & 0 \end{bmatrix}. \quad (2.72)$$

Setting $\bar{\omega} = [\omega_1 \ \omega_2 \ \dots \ \omega_K] \in \mathbb{R}^{3 \times K}$, the rotation matrix from t to $t + K\tau_s$ can be approximated by the outcome of the following function [10] [23]

$$R_g(\bar{\omega}) = (I + \tau_s \Omega(\bar{\omega}e_1))(I + \tau_s \Omega(\bar{\omega}e_2)) \dots (I + \tau_s \Omega(\bar{\omega}e_K)) \quad (2.73)$$

where e_k is the k^{th} normal vector in \mathbb{R}^K .

2.7.3 Calibration Algorithm

Assume that the sensor's platform rotates N times with a short period of stillness between them. During every rotation, the gyroscope is regularly sampled every τ_s seconds and the samples are recorded. Recording begins from the still position, just before the rotation begins, and ends at the next still position, just after the rotation ends. We also assume that every one of the three gyroscope's axes is rotated significantly in at least one of the rotations.

Let ω_j^n be the j^{th} gyroscope's sample measured sample (i.e. y_g in (2.6)) during the n^{th} rotation, $n = 1, 2, \dots, N$. Using (2.6), we get the calibrated sample

$$\tilde{\omega}_j^n = H_g(\omega_j^n - h_g)$$

where $H_g = T_g^{-1}$. Here we assume that T_g is invertible. This is a rational assumption as a non-invertible T_g would imply that not all three axes are expressed in the output of the sensor. For every rotation, we form the matrix

$$\bar{\omega}^n = [\tilde{\omega}_1^n \ \tilde{\omega}_2^n \ \dots \ \tilde{\omega}_{M_n}^n]$$

where M_n is the number of the recorded samples during the n^{th} rotation. Then, using (2.73), for every rotation, we derive a rotation matrix as a function of the calibration parameters H_g and h_g in (2.6).

$$R_g^n = R_g(\bar{\omega}^n) \quad (2.74)$$

Calculating the rotations using accelerometer's and magnetometer's measurements, as in Section 2.7.1, requires measurements of both sensors before and after every rotation, while the

sensors are still. Let f_{begin}^n and f_{end}^n be the measured specific force vectors exactly before and after rotation n , $n = 1, 2, \dots, N$. Similarly let m_{begin}^n and m_{end}^n be the corresponding vectors of the magnetic field. To minimize the effect of the sensors' noise, we prefer to define the above four vectors as the average of L samples. Then, using (2.71), for every rotation, we derive the accelerometer-magnetometer rotation matrix

$$R_{am}^n = R_{am}(f_{begin}^n, f_{end}^n, m_{begin}^n, m_{end}^n). \quad (2.75)$$

To calibrate the gyroscope we minimize the mean square error between the rotation calculated using gyroscope's measurements R_g^n and the corresponding rotation calculated using accelerometer's and magnetometer's measurements R_{am}^n . This is done for all N rotations simultaneously, and so the calibration procedure can be posed as the minimization of cost function $J(x)$, where

$$J(x) = \sum_{n=1}^N \left\{ \|R_{am}^n - R_g^n\|^2 \right\} + \lambda \|H_g(\omega_{still} - h_g)\|^2 \quad (2.76)$$

and

$$x = \left[\text{vec}(H_g)^T \quad h_g^T \right]^T$$

The positive constant λ is selected to balance the contribution of the two summands in (2.76) where the second one is for nulling the sensor's bias. Note that ω_{still} is the gyroscope's output (y_g in (2.6)) when it is still (i.e. $\omega = 0$ in (2.6)), also defined as the average of M measurements to reduce random noise.

We solve (2.76) using the gradient descent method with the gradient of $J(x)$ be numerically calculated. Assuming small scale-factor, cross-coupling and bias errors we initialize H_g to the 3×3 identity matrix and h_g to the 3×1 zero vector. Gyroscope calibration algorithm is shown in Algorithm 4, where a and b are positive numbers for the line search.

Algorithm 4: Gyroscope Calibration

Step 1: Initialize $H_g = I_3$, $h_g = 0_{3 \times 1}$,

Step 2: Initialize t, a and b

Step 3: Calculate the gradient:

$$\Delta x = -\nabla J(x)$$

Step 4: Choose step size:

$$\text{while } J(x + t\Delta x) > J(x) + at\nabla J(x)^T \Delta x$$

$$t := \beta t$$

Step 5: Update $x = x + \Delta x$

Step 6: Calculate $J(x)$

Step 7: Repeat steps 3-6 until $J(x)$ is sufficiently small

2.7.4 Algorithm's Evaluation

Similar to the case of the accelerometer and the magnetometer, several different datasets were used to evaluate the performance of the proposed algorithm. In Figure 2.11, the convergence of the proposed algorithm using five different datasets and the calibration procedure presented in [25] is shown.

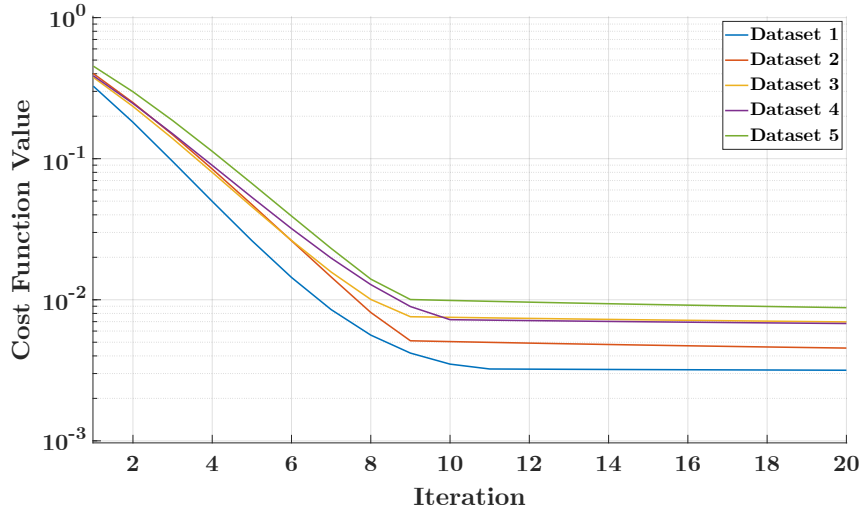


Figure 2.11: Convergence of gyroscope calibration algorithm

As seen in Figure 2.11 gyroscope calibration algorithm converges after only a few iterations for all five datasets, while the corresponding cost function appears to be monotonic.

To demonstrate the repeatability of the gyroscope's calibration algorithm, we use the distances between the derived offset vectors and calibration matrices using different datasets defined in Section 2.5.2. The calculated distances are shown below

$$\begin{aligned} \bar{d}(h_{g_1}, h_{g_2}, \dots, h_{g_5}) &= 0.0480 \\ \bar{D}(T_{g_1}, T_{g_2}, \dots, T_{g_5}) &= 0.0222 \end{aligned} \tag{2.77}$$

The small values of the normalized average distances (2.77) indicate good repeatability of the proposed calibration algorithm i.e. the algorithm return similar calibration parameters for different dataset inputs.

Finally, to assess the performance of the gyroscope calibration algorithm, we rotated the device by 90° , about it's x-axis five times. The corresponding Euler angle derived from the measurements for each rotation is presented in table 2.4.

Rotation	Roll Angle
1	90.348°
2	89.769°
3	90.216°
4	90.402°
5	89.976°

Table 2.4: Measured roll angle for 90° rotation about x-axis.

The measured Euler angles in Table 2.4 are very close to the true rotation angle (90°) indicating the good accuracy of the proposed calibration algorithm without using any special piece of equipment and without any external attitude reference.

References

- [1] R. Alonso and Shuster. “TWOSTEP: A fast robust algorithm for attitude-independent magnetometer-bias determination.” In: *The Journal of the Astronautical Sciences*. 50 (2002), pp. 433–451.
- [2] N. Ammann, A. Derksen, and C. Heck. “A novel magnetometer-accelerometer calibration based on a least squares approach”. In: *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. June 2015, pp. 577–585. DOI: 10.1109/ICUAS.2015.7152338.
- [3] H. D. Black. “A passive system for determining the attitude of a satellite”. In: *AIAA Journal* 2.7 (July 1964), pp. 1350–1351. ISSN: 0001-1452. DOI: 10.2514/3.2555. URL: <https://doi.org/10.2514/3.2555>.
- [4] C. Chu. “Video stabilization for stereoscopic 3D on 3D mobile devices”. In: *2014 IEEE International Conference on Multimedia and Expo (ICME)*. July 2014, pp. 1–6. DOI: 10.1109/ICME.2014.6890150.
- [5] E. Dorveaux et al. “Iterative calibration method for inertial and magnetic sensors”. In: *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. Dec. 2009, pp. 8296–8303. DOI: 10.1109/CDC.2009.5399503.
- [6] C. Fischer, P. Talkad Sukumar, and M. Hazas. “Tutorial: Implementing a Pedestrian Tracker Using Inertial Sensors”. In: *IEEE Pervasive Computing* 12.2 (Apr. 2013), pp. 17–27. ISSN: 1536-1268. DOI: 10.1109/MPRV.2012.16.
- [7] I. Frosio, F. Pedersini, and N. A. Borghese. “Autocalibration of MEMS Accelerometers”. In: *IEEE Transactions on Instrumentation and Measurement* 58.6 (June 2009), pp. 2034–2041. ISSN: 0018-9456. DOI: 10.1109/TIM.2008.2006137.

- [8] M. V. Gheorghe. “Calibration for tilt-compensated electronic compasses with alignment between the magnetometer and accelerometer sensor reference frames”. In: *2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. May 2017, pp. 1–6. DOI: 10.1109/I2MTC.2017.7969813.
- [9] M. Glueck, A. Buhmann, and Y. Manoli. “Autocalibration of MEMS accelerometers”. In: *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*. May 2012, pp. 1788–1793. DOI: 10.1109/I2MTC.2012.6229157.
- [10] Paul D. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2013.
- [11] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2013.
- [12] M. Kok et al. “Calibration of a magnetometer in combination with inertial sensors”. In: *2012 15th International Conference on Information Fusion*. July 2012, pp. 787–793.
- [13] Manon Kok and Thomas B. Schon. “Magnetometer Calibration Using Inertial Sensors”. In: *IEEE Sensors Journal* 16.14 (July 2016), pp. 5679–5689. ISSN: 1530-437X. DOI: 10.1109/JSEN.2016.2569160.
- [14] Manon Kok and Thomas B. Schön. “Maximum likelihood calibration of a magnetometer using inertial sensors”. In: *IFAC Proceedings Volumes* 47.3 (2014). 19th IFAC World Congress, pp. 92–97. ISSN: 1474-6670. DOI: <https://doi.org/10.3182/20140824-6-ZA-1003.02025>.
- [15] Manon Kok and Thomas B. Schön. “Maximum likelihood calibration of a magnetometer using inertial sensors”. In: *IFAC Proceedings Volumes* 47.3 (2014). 19th IFAC World Congress, pp. 92–97. ISSN: 1474-6670. DOI: <https://doi.org/10.3182/20140824-6-ZA-1003.02025>. URL: <http://www.sciencedirect.com/science/article/pii/S1474667016415983>.
- [16] Wei Li and Jinling Wang. “Effective Adaptive Kalman Filter for MEMS-IMU/Magnetometers Integrated Attitude and Heading Reference Systems”. In: *Journal of Navigation* 66.1 (2013), pp. 99–113. DOI: 10.1017/S0373463312000331.
- [17] Xiang Li and Zhi Li. “A new calibration method for tri-axial field sensors in strap-down navigation systems”. In: *Measurement Science and Technology* 23.10 (Sept. 2012), p. 105105. DOI: 10.1088/0957-0233/23/10/105105.
- [18] Z. Li et al. “An Improved Pedestrian Navigation System Using IMU and Magnetometer”. In: *2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)*. Dec. 2017, pp. 1639–1642. DOI: 10.1109/ICCSEC.2017.8446992.

- [19] X. Lu, Z. Liu, and J. He. “Maximum Likelihood Approach for Low-Cost MEMS Triaxial Accelerometer Calibration”. In: *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. Vol. 01. Aug. 2016, pp. 179–182. DOI: 10.1109/IHMSC.2016.184.
- [20] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. “Estimation of IMU and MARG orientation using a gradient descent algorithm”. In: *2011 IEEE International Conference on Rehabilitation Robotics*. June 2011, pp. 1–7. DOI: 10.1109/ICORR.2011.5975346.
- [21] J Metge et al. “Calibration of an inertial-magnetic measurement unit without external equipment, in the presence of dynamic magnetic disturbances”. In: *Measurement Science and Technology* 25.12 (2014), p. 125106. URL: <http://stacks.iop.org/0957-0233/25/i=12/a=125106>.
- [22] S. Nadig et al. “In-run scale factor and drift calibration of MEMS gyroscopes with rejection of acceleration sensitivities”. In: *2016 IEEE International Symposium on Inertial Sensors and Systems*. Feb. 2016, pp. 144–145. DOI: 10.1109/ISISS.2016.7435569.
- [23] Aboelmagd Noureldin, Tashfeen B. Karamat, and Jacques Georgy. *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*. Springer-Verlag Berlin Heidelberg, 2013.
- [24] K. Papafotis and P. P. Sotiriadis. “Accelerometer and Magnetometer Joint Calibration and Axes Alignment”. In: *Technologies* 8(1), 11 (Jan. 2020). DOI: 10.3390/technologies8010011.
- [25] K. Papafotis and P. P. Sotiriadis. “MAG.I.C.AL.—A Unified Methodology for Magnetic and Inertial Sensors Calibration and Alignment”. In: *IEEE Sensors Journal* 19.18 (Sept. 2019), pp. 8241–8251. ISSN: 1530-437X. DOI: 10.1109/JSEN.2019.2919179.
- [26] U. Qureshi and F. Golnaraghi. “An Algorithm for the In-Field Calibration of a MEMS IMU”. In: *IEEE Sensors Journal* 17.22 (Nov. 2017), pp. 7479–7486. ISSN: 1530-437X. DOI: 10.1109/JSEN.2017.2751572.
- [27] P. Schonemann. “A generalized solution of the orthogonal procrustes problem”. In: *Psychometrika* 31.1 (Mar. 1966), pp. 1–10.
- [28] Tianjiao Shi et al. “Fall Detection Algorithm Based on Triaxial Accelerometer and Magnetometer.” In: *Engineering Letters* 24.2 (2016).
- [29] M. D. Shuster and S. D. Oh. “Three-axis attitude determination from vector observations”. In: *Journal of Guidance, Control, and Dynamics* 4.1 (Jan. 1981), pp. 70–77. ISSN: 0731-5090. DOI: 10.2514/3.19717. URL: <https://doi.org/10.2514/3.19717>.
- [30] P. P. Sotiriadis and K. Papafotis. “A Single-Step Method for Accelerometer and Magnetometer Axes Alignment”. In: *IEEE Transactions on Instrumentation and Measurement* (2020), pp. 1–1.

- [31] Ross Stirling et al. "An innovative shoe-mounted pedestrian navigation system". In: Jan. 2003, pp. 110–5.
- [32] Gilbert Strang. *Linear Algebra and Its Applications*. Brooks Cole/Cengage Learning, 2007.
- [33] Z F Syed et al. "A new multi-position calibration method for MEMS inertial navigation systems". In: *Measurement Science and Technology* 18.7 (2007), p. 1897. URL: <http://stacks.iop.org/0957-0233/18/i=7/a=016>.
- [34] Q. Tang et al. "An improved scale factor calibration model of MEMS gyroscopes". In: *2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*. May 2014, pp. 752–755. DOI: 10.1109/I2MTC.2014.6860843.
- [35] J. F. Vasconcelos et al. "Geometric Approach to Strapdown Magnetometer Calibration in Sensor Frame". In: *IEEE Transactions on Aerospace and Electronic Systems* 47.2 (Apr. 2011), pp. 1293–1306. ISSN: 0018-9251. DOI: 10.1109/TAES.2011.5751259.
- [36] Y. Wu and S. Luo. "On Misalignment Between Magnetometer and Inertial Sensors". In: *IEEE Sensors Journal* 16.16 (Aug. 2016), pp. 6288–6297. ISSN: 1530-437X. DOI: 10.1109/JSEN.2016.2582751.
- [37] Y. Wu and L. Pei. "Gyroscope Calibration via Magnetometer". In: *IEEE Sensors Journal* 17.16 (Aug. 2017), pp. 5269–5275. ISSN: 1530-437X. DOI: 10.1109/JSEN.2017.2720756.
- [38] Y. Wu and W. Shi. "On Calibration of Three-Axis Magnetometer". In: *IEEE Sensors Journal* 15.11 (Nov. 2015), pp. 6424–6431. ISSN: 1530-437X. DOI: 10.1109/JSEN.2015.2459767.
- [39] Y. Wu et al. "Dynamic Magnetometer Calibration and Alignment to Inertial Sensors by Kalman Filtering". In: *IEEE Transactions on Control Systems Technology* 26.2 (Mar. 2018), pp. 716–723. ISSN: 1063-6536. DOI: 10.1109/TCST.2017.2670527.
- [40] Hongliang Zhang et al. "Improved multi-position calibration for inertial measurement units". In: *Measurement Science and Technology* 21.1 (2010), p. 015107. URL: <http://stacks.iop.org/0957-0233/21/i=1/a=015107>.
- [41] J. Zhang et al. "A novel scale factor calibration method for a MEMS gyroscope based on virtual coriolis force". In: *10th IEEE International Conference on Nano/Micro Engineered and Molecular Systems*. Apr. 2015, pp. 58–62. DOI: 10.1109/NEMS.2015.7147373.
- [42] Y. Zhong et al. "A New Drone Accelerometer Calibration Method". In: *2018 37th Chinese Control Conference (CCC)*. July 2018, pp. 9928–9933. DOI: 10.23919/ChiCC.2018.8482568.

3

Magnetic Field Sensors' Calibration: Algorithms' Overview and Comparison

The calibration of three-axis magnetic field sensors is reviewed. Seven representative algorithms dealing with the in-situ calibration of magnetic field sensors without requiring any special piece of equipment are reviewed. The algorithms are presented in a user friendly, directly applicable step-by-step form, and are compared in terms of accuracy, computational efficiency, and robustness using both real sensors' data and artificial data with a known sensor measurement distortion.

3.1 Introduction

Magnetic field sensors (magnetometers) are nowadays widely used in a plethora of commercial, industrial, marine, aerospace and military applications. Their applications include but not limited to navigation and attitude estimation, geophysical surveys, archaeology, entertainment devices, consumer electronics and others.

In most applications, sensor's calibration is essential in order to achieve the desirable accuracy level. The purpose of magnetic field sensors' calibration is a twofold. First, as in the case of every measurement unit, calibration ensures that the measurement of the standalone sensor corresponds to the actual value of the magnetic field. To do so, calibration must compensate for all static (manufacturing imperfections etc.) and active (temperature, humidity, etc.) phenomena effecting the accuracy of the sensor's measurement. In addition, when a magnetic sensor is embedded in a larger system, other components of the system may cause disturbances (both static and active ones) to the local magnetic field. Static disturbances are usually caused by magnetic and ferromagnetic materials in the vicinity of the sensor; called hard-iron distortion and soft-iron distortion respectively (more information are given in Section 2). Mechanical or electronic structures embedded in the system, such as motors and coils could also actively distort the local magnetic field and cause significant measurement error.

This review paper focuses on algorithms correcting the dominant linear time-invariant (static) measurement errors, requiring no special piece of equipment for their application. Such algorithms

are most commonly used for in-situ calibration of magnetic field sensors which are usually in chip form and embedded in larger systems. The paper presents seven representative calibration algorithms for three-axis magnetometers and compares them in terms of accuracy, robustness, computational efficiency and ease of deployment. The seven algorithms are briefly presented, to introduce all required mathematical expressions, and are summarized in an easy-to-develop, step-by-step form. For the details of the algorithms, the reader is referred to the original works.

The selection of the particular algorithms was done based on their popularity and on our attempt to present as many different calibration approaches as possible. The TWOSTEP [2] algorithm is one of the first algorithms that addressed the full calibration problem (and probably the most popular one). At a later time, Elkaim and Vasconcelos [30] proposed a geometric approach of TWOSTEP which is also very popular. At the same time, Dorveaux et al. [7] offered a nonlinear formulation of the problem and they treated it in an innovative, strictly iterative way. In addition, Wu and Shi [31] suggested the most complete formulation of the calibration problem as an optimal maximum likelihood estimation one. The TWOSTEP algorithm, as well as the algorithms proposed by Vasconcelos et al. and Wu et al., consist of a first step deriving an initial solution, and, a second step for improving it. On the other hand, Papafotis and Sotiriadis [25] recommended an iterative approach based on a twofold minimization, which was shown to be extremely effective. Furthermore, a real-time approach by Crassidis et al. [5] using the popular Kalman Filter is discussed. Finally, to represent the recent trends towards Machine Learning, an AI method applying Particle Swarm Optimization on the estimation problem is explored [1].

Please note that this review focuses on works for in-situ calibration of three-axis magnetic field sensor without using any special piece of equipment or any other additional sensor. Thus, several interesting works dealing with magnetometer's calibration, in combination with inertial sensors, [24, 18, 32, 19, 21] are not included in this work.

The rest of the paper is organized as follows. First, a standard error model for three-axis magnetic field sensors is presented in Section 2. In Section 3-9, seven representative algorithms are discussed in chronological order of publication. In section 10, a method for generating artificial data is proposed and algorithms are evaluated via extensive Monte Carlo simulation to identify their performance. In addition, the algorithms are evaluated using several real sensor's measurements in order to evaluate their performance under real-world conditions. Finally, section 11 summarizes our findings and provides brief comments for each algorithm. The notation used along the paper is presented in Table 3.1.

3.2 Magnetic Field Sensor's Error Sources and Measurement Model

In this section, the most important linear, time-invariant error sources of three-axis magnetic field sensors are presented. Based on them, a mathematical model relating the sensor's measurement with the actual value of the magnetic field is derived.

$\ \cdot\ $	Euclidean Norm
$\ \cdot\ _F$	Frobenius Norm
$\text{vec}(\cdot)$	Vectorization of Matrix
$\text{diag}(\cdot)$	Diagonal Matrix
$\text{chol}(\cdot)$	Cholesky Factorization
$I_{n \times n}$	$n \times n$ Identity Matrix
$0_{m \times 1}$	$m \times 1$ Zero Vector
\mathcal{N}	Normal Distribution
\mathcal{U}	Uniform Distribution
∇	Gradient Vector
∇^2	Hessian Matrix
\otimes	Kronecker Product
$O(3)$	Orthogonal Group of dimension 3
$SO(3)$	3D Rotation Group
$\mathcal{U}(3)$	Group of 3×3 Upper Triangular Matrices

Table 3.1: Notation

The total output error of a magnetic sensor is a combination of several error sources related to the sensing element itself, the instrumentation electronics, manufacturing imperfections and distortions caused by magnetic and ferromagnetic materials in the vicinity of the sensor. The linear, time-invariant error sources with the most significant contribution in the total sensor's error, are listed below:

- **Bias, or offset**; all magnetic sensors suffer from bias, which is a constant distortion. In many cases, it is the most important defect in the sensor's overall error. A 3×1 vector, h_s , is used to model it.
- **Scale-factor** error represents the input-output gain error of the sensor. It is modeled by a 3×3 diagonal matrix, T_{sf} .
- **Cross-coupling or Non-Orthogonality** inaccuracies are resulted by the non-ideal alignment of the sensor's axes during manufacturing and are modeled by a 3×3 matrix, T_{cc} .
- **Soft-iron distortion** is caused by ferromagnetic materials in the vicinity of the sensor, attached to the sensor's coordinate frame. Those materials do not generate their own magnetic field, but instead alter the existing magnetic field locally, resulting in a measurement discrepancy. This effect is modeled by a 3×3 matrix, T_{si} .
- **Hard-iron distortion** is due to magnetic materials attached to the sensor's coordinate frame. As a consequence of the persistent magnetic field created by those materials, the sensor's output has a constant bias. Hard-iron distortion is modeled by a 3×1 vector, h_{hi} .

- **Random noise** is the stochastic error in the sensor's output. It is induced by the sensor's mechanical and electrical architecture. It is modeled by a 3×1 vector, ε , and it is most commonly assumed to be a sequence of white noise, i.e. $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.

Let m be the 3×1 true magnetic field vector and y be the 3×1 measurement vector. With the aforementioned error terms in mind, a widely accepted and well-referenced measurement model for a three-axis magnetometer is the following [25, 2, 31, 30, 1, 4, 5, 13]

$$y = T_{sf}T_{cc}(T_{si}m + h_{hi}) + h_s + \varepsilon \quad (3.1)$$

In most applications, the exact contribution of each error term in (3.1) is of no concern and thus, instead of (3.1), most calibration algorithms use the following, compact form of (3.1)

$$y = Tm + h + \varepsilon \quad (3.2)$$

where $T \triangleq T_{sf}T_{cc}T_{si}$ and $h \triangleq T_{sf}T_{cc}h_{hi} + h_s$.

This work focuses on algorithms intended to be used with magnetic field sensors requiring no special piece of equipment. In such cases, the calibration is done in the sensor's (body) coordinate frame implying that both the measurement vector, y and the true magnetic field vector, m in (3.2) are expressed in the sensor's coordinate frame.

Note that when expensive laboratory equipment is not available, both the calibration parameters T and h in (3.2), and the magnetic field vector, m , are unknown. Thus, in most works, multiple measurements of the local (Earth's) magnetic field are used to derive T and h . Note that the Earth's magnetic field varies with location and time and its value (magnitude and direction) is only approximately known by magnetic models such as International Geomagnetic Reference Field model (IGRF) [15]. However it is reasonable to assume that the magnitude of the magnetic field is (locally) constant during the calibration procedure. Based on this fact, most authors formulate an optimization or an estimation problem to derive T and h .

3.3 Alonso and Shuster (TWOSTEP) [2]

The TWOSTEP algorithm consists of an analytic centering approach [10, 20] for its first step, while in the second step the solution is optimized numerically. The authors initially solved the problem of bias, h , determination when attitude is not known [3] and then extended their method to determine matrix T as well [2].

It is motivated by the assumption that matrix T should not be far from a pure rotation. Therefore by applying polar decomposition it can be written as $T = (I_{3 \times 3} + D)^{-1}O$ where O is an orthogonal matrix and D is a symmetric 3×3 matrix so as $(I_{3 \times 3} + D)^{-1}$ to be positive definite. Matrix O can be integrated into vector m since it does not alter its norm. The equivalent measurement model is

$$y = \hat{T}\hat{m} + h + \varepsilon \quad (3.3)$$

where

$$\begin{aligned}\hat{T} &\triangleq (I_{3 \times 3} + D)^{-1} \\ \hat{m} &\triangleq Om\end{aligned}$$

Therefore, for the full calibration, D and h must be estimated. To this purpose, a set of N measurements, y_k , $k = 1, 2, \dots, N$, is used and the corresponding effective measurements z_k , $k = 1, 2, \dots, N$, are defined as

$$\begin{aligned}z_k &\triangleq \|y_k\|^2 - \|\hat{m}_k\|^2 \\ &= \|y_k\|^2 - \|m_k\|^2.\end{aligned}\tag{3.4}$$

The last ones can be decomposed into a deterministic part plus an approximately Gaussian noise term, v_k with mean μ_k and variance σ_k^2 , i.e. $v_k \sim \mathcal{N}(\mu_k, \sigma_k^2)$, given by

$$\mu_k = -3\sigma^2\tag{3.5\alpha'}$$

$$\sigma_k^2 = 4\sigma^2((I_{3 \times 3} + D)y_k - h)^T((I_{3 \times 3} + D)y_k - h) + 6\sigma^4\tag{3.5\beta'}$$

Since D and h are unknown, the variance σ_k^2 is assumed to be similar to measurement's output error variance σ^2 . Hence μ_k and σ_k^2 can be assumed independent of k .

To estimate D and h , Alonso and Shuster define the auxiliary quantities

$$E \triangleq D^2 + 2D\tag{3.6\alpha'}$$

$$c \triangleq (I + D)h\tag{3.6\beta'}$$

and the estimation vector θ' which contains the elements of the 3×1 vector c and the elements of the 3×3 symmetric matrix E and is structured as follows

$$\theta' = [c^T E_{11} E_{22} E_{33} E_{12} E_{13} E_{23}]^T.\tag{3.7}$$

TWOSTEP algorithm functions on the estimation vector θ' and thus on the auxiliary parameters, E and c and not on the actual calibration parameters, D and h . The transformation from E and c to D and h is described in (3.15)-(3.17).

3.3.1 Initial Estimate

For every measurement, y_k , $k = 1, 2, \dots, N$, a corresponding auxiliary variable is defined

$$S_k = [y_{k,1}^2 \ y_{k,2}^2 \ y_{k,3}^2 \ 2y_{k,1}y_{k,2} \ 2y_{k,1}y_{k,3} \ 2y_{k,2}y_{k,3}]\tag{3.8\alpha'}$$

$$L_k = [2y_k^T \ | \ -S_k]\tag{3.8\beta'}$$

The centering approximation is done using the following weighted averages

$$\bar{z} \triangleq \bar{\sigma}^2 \sum_{k=1}^N \frac{1}{\sigma_k^2} z_k \quad \bar{L} \triangleq \bar{\sigma}^2 \sum_{k=1}^N \frac{1}{\sigma_k^2} L_k \quad \bar{\mu} \triangleq \bar{\sigma}^2 \sum_{k=1}^N \frac{1}{\sigma_k^2} \mu_k \quad (3.9)$$

along with the corresponding centered values

$$\tilde{z}_k = z_k - \bar{z} \quad \tilde{L}_k = L_k - \bar{L} \quad \tilde{\mu}_k = \mu_k - \bar{\mu} \quad (3.10)$$

where

$$\bar{\sigma}^2 \triangleq \left(\sum_{k=1}^N \frac{1}{\sigma_k^2} \right)^{-1}$$

The centered estimation for θ' is given by

$$\tilde{\theta}' = \tilde{P}_{\theta'\theta'} \sum_{k=1}^N \frac{1}{\sigma_k^2} (\tilde{z}_k - \tilde{\mu}_k) \tilde{L}_k^T \quad (3.11\alpha')$$

$$\tilde{P}_{\theta'\theta'}^{-1} = \tilde{F}_{\theta'\theta'} = \sum_{k=1}^N \frac{1}{\sigma_k^2} \tilde{L}_k^T \tilde{L}_k \quad (3.11\beta')$$

with $\tilde{P}_{\theta'\theta'}$ denoting the centered covariance matrix and $\tilde{F}_{\theta'\theta'}$ denoting the centered Fischer information matrix.

3.3.2 Solution Improvement Step

The second step improves the previous estimate of vector θ , derived in (3.11), via Gauss-Newton method using the centered estimate $\tilde{\theta}'$ as the initial guess. The estimation is updated as follows

$$\theta'_{i+1} = \theta'_i - \left[\tilde{F}_{\theta'\theta'} + \frac{1}{\bar{\sigma}^2} (\bar{L} - \phi(\theta'_i))^T (\bar{L} - \phi(\theta'_i)) \right]^{-1} g(\theta'_i) \quad (3.12)$$

where

$$v = (I_{3 \times 3} + E)^{-1} c \quad (3.13\alpha')$$

$$\phi(\theta') = [2v^T \quad -v_1^2 \quad -v_2^2 \quad -v_3^2 \quad -2v_1v_2 \quad -2v_1v_3 \quad -2v_2v_3] \quad (3.13\beta')$$

$$g(\theta') = \tilde{P}_{\theta'\theta'}^{-1} (\theta' - \tilde{\theta}') - \frac{1}{\bar{\sigma}^2} (\bar{z} - \bar{L}\theta' + c^T v - \bar{\mu}) (\bar{L}^T - \phi(\theta')) \quad (3.13\gamma')$$

with v_j denoting the j^{th} element of vector v . At every iteration the 3×3 symmetric matrix E and the 3×1 vector c are updated according to the current estimation vector θ'_i using (3.7).

Alonso and Shuster define the following quantity in order to establish a stop condition for the Gauss-Newton method.

$$\eta_i \triangleq (\theta'_{i+1} - \theta'_i)^T \left[\tilde{F}_{\theta'\theta'} + \frac{1}{\bar{\sigma}^2} (\bar{L} - \phi(\theta'_i))^T (\bar{L} - \phi(\theta'_i)) \right] (\theta'_{i+1} - \theta'_i) \quad (3.14)$$

The iterations continue until η_i became smaller than a predetermined threshold.

After sufficient iterations, an optimal estimation of matrix E^* and of vector c^* is derived (3.7). The derived solution is then transformed to the quantities D^* and h^* . By applying SVD [29] on the symmetric matrix E^* , it is written as

$$E^* = USU^T \quad (3.15)$$

where $S = \text{diag}(s_1, s_2, s_3)$, $U \in O(3)$. Advancing, we find the diagonal matrix $W = \text{diag}(w_1, w_2, w_3)$ that satisfies $S = 2W + W^2$. Typically, the elements of S are much smaller than unity [2] so that a solution exists. The diagonal elements of W are given by

$$w_j = -1 + \sqrt{1 + s_j} \quad (3.16)$$

for $j = 1, 2, 3$. The estimates of the matrix D^* and bias vector h^* are then given by

$$D^* = UWU^T \quad (3.17\alpha')$$

$$h^* = (I_{3 \times 3} + D^*)^{-1} c^* \quad (3.17\beta')$$

and are related to the calibration parameters T and h of the measurement model (3.2) as follows

$$T = (I_{3 \times 3} + D^*)^{-1} \text{ and } h = h^* \quad (3.18)$$

Algorithm 5: Alonso and Shuster (TWOSTEP) [2]

Step 1: Calculate z_k, L_k , for $k = 1, 2, \dots, N$
by using (3.4), (3.8)

Step 2: Calculate the centered values \tilde{z}_k, \tilde{L}_k for $k = 1, 2, \dots, N$ (3.9), (3.10)

Step 3: Calculate centered estimate $\tilde{\theta}'$ and covariance matrix $\tilde{P}_{\theta'}$ (3.11)

Step 4: Extract c and E from θ' (3.7)

Step 5: Calculate $\phi(\theta')$ and $g(\theta')$ (3.13)

Step 6: Update θ' (3.12)

Step 7: Calculate η (3.14)

Step 8: Repeat steps 4-7 until η is sufficiently small
or maximum iterations are met

Step 9: Apply SVD on E^* (3.15) and define matrix W (3.16)

Step 10: Calculate D^*, h^* (3.17) and T, h (3.18)

3.4 Crassidis et al. [5]

The authors of [5] realized that real-time applications demand real-time calibration methods. To this end, based on TWOSTEP [2], Crassidis et al. formulate a real-time estimation problem for the derivation of the calibration parameters T and h and solve it using the extended Kalman Filter approach [6]

Following TWOSTEP, a bias vector h and for symmetric matrix D is desired. The estimation vector θ contains the elements of h and D and is structured as it follows

$$\theta = [h^T \ D_{11} \ D_{22} \ D_{33} \ D_{12} \ D_{13} \ D_{23}]^T \quad (3.19)$$

Because the vector θ is constant, the state model is given by $\dot{\theta} = 0$. The effective measurement is given by $z_k = \|y_k\|^2 - \|m_k\|^2$ (3.4) while the measurement's model is given by $z_k = \phi(\theta_k) + v_k$ where

$$\phi(\theta_k) = -y_k^T (2D_k + D_k^2) y_k + 2y_k^T (I_{3 \times 3} + D_k) h_k - \|h_k\|^2 \quad (3.20)$$

and effective measurement's noise $v_k \sim \mathcal{N}(\mu_k, \sigma_k^2)$ follows (3.5). At each iteration D_k and h_k are extracted from θ_k according to (3.19). The propagation is as it follows

$$\theta_{k+1} = \theta_k + K_k [z_k - \phi(\theta_k)] \quad (3.21\alpha')$$

$$P_{k+1} = [I_{9 \times 9} - K_k H(\theta_k)] P_k \quad (3.21\beta')$$

$$K_k = P_k H^T(\theta_k) [H(\theta_k) P_k H^T(\theta_k) + \sigma_k^2]^{-1} \quad (3.21\gamma')$$

where P_k is the covariance of the estimated parameters for h and D at step k . The matrix $H(\theta_k)$ is the linearization matrix of $\phi(\theta_k)$ and is defined as

$$H(\theta_k) = [2y_k^T (I_{3 \times 3} + D_k) - 2h_k^T \quad -S_k F_k + 2J_k] \quad (3.22)$$

where

$$S_k = [y_{k,1}^2 \ y_{k,2}^2 \ y_{k,3}^2 \ 2y_{k,1}y_{k,2} \ 2y_{k,1}y_{k,3} \ 2y_{k,2}y_{k,3}] \quad (3.23\alpha')$$

$$J_k = [y_{k,1}h_{k,1} \ y_{k,2}h_{k,2} \ y_{k,3}h_{k,3} \ y_{k,1}h_{k,2} + y_{k,2}h_{k,1} \quad (3.23\beta')$$

$$y_{k,1}h_{k,3} + y_{k,3}h_{k,1} \ y_{k,2}h_{k,3} + h_{k,3}h_{k,2}]$$

$$F_k = \begin{bmatrix} \Delta_1 & 0 & 0 & 2D_{k,12} & 2D_{k,13} & 0 \\ 0 & \Delta_2 & 0 & 2D_{k,12} & 0 & 2D_{k,23} \\ 0 & 0 & \Delta_3 & 0 & 2D_{k,13} & 2D_{k,23} \\ D_{k,12} & D_{k,12} & 0 & \Delta_4 & D_{k,23} & D_{k,13} \\ D_{k,13} & 0 & D_{k,13} & D_{k,23} & \Delta_5 & D_{k,12} \\ 0 & D_{k,23} & D_{k,23} & D_{k,13} & D_{k,12} & \Delta_6 \end{bmatrix} \quad (3.23\gamma')$$

where

$$\begin{aligned}
\Delta_1 &= 2(1 + D_{k,11}) \\
\Delta_2 &= 2(1 + D_{k,22}) \\
\Delta_3 &= 2(1 + D_{k,33}) \\
\Delta_4 &= 2 + D_{k,11} + D_{k,22} \\
\Delta_5 &= 2 + D_{k,11} + D_{k,33} \\
\Delta_6 &= 2 + D_{k,22} + D_{k,33}
\end{aligned} \tag{3.24}$$

The choice of σ_k^2 , again, seems to be unimportant. Thus it can be assumed similar to σ^2 , likewise with TWOSTEP. Given a set of N measurements, the EKF will provide an optimal estimation vector $\theta^* = \theta_N$ from which an optimal vector $h^* = h_N$ and a matrix $D^* = D_N$ can be extracted according to (3.19). Therefore, the full calibration parameters (3.2) are given below

$$T = (I_{3 \times 3} + D^*)^{-1} \text{ and } h = h^* \tag{3.25}$$

Even though the authors focused on sequential calibration methods, they do not seem to offer any advantage over batch algorithms for static environments due to the fact that the estimated variables are slowly varying. The authors verified the robustness of their method via simulations assuming either white noise or coloured noise, for a long-duration spacecraft's on-orbit flight. This method makes use of the concept of effective measurement (3.4) likewise with TWOSTEP. Therefore, similar assumptions and consumption could be made.

Algorithm 6: Crassidis et al. (Kalman Filter) [5]

Step 1: Initialize θ and $k = 0$

Step 2: **for each** measurement do:

Calculate z_k (3.4)

Extract D_k and h_k from θ_k (3.19)

Calculate S_k, J_k, F_k (3.23) and $H(\theta_k)$ (3.22)

Calculate Kalman Gain K_k (3.21)

Update estimation: $\theta_k \leftarrow \theta_{k+1}$

Update covariance matrix: $P_k \leftarrow P_{k+1}$ (3.21)

$k \leftarrow k + 1$

Step 3: Extract D^* and h^* from θ^* (3.19)

Step 4: Calculate T and h (3.25)

3.5 Dorveaux et al. [7]

An iterative algorithm for the calibration of magnetic field sensors based on iterations of a least-squares problem is introduced in [7]. In the beginning of the algorithm, the measurements lie on an ellipsoid according to (3.2). In each iteration, the measurements move from the initial ellipsoid to the unit sphere, following a cost function minimization algorithm.

The authors in [7] use the following variation of the measurement model of (3.2)

$$m = Ay + B \quad (3.26)$$

where $A = T^{-1}$, $B = -T^{-1}h$ and the measurement noise, ε , is neglected.

The algorithm begins by considering an initial estimate of the magnetic field vectors, denoted by $\tilde{m}_k(0)$ and defined as

$$\tilde{m}_k(0) = y_k, \quad k = 1, 2, \dots, K \quad (3.27)$$

In every iteration, the following cost function is formulated and minimized using the least squares method.

$$J(A, B, n) = \sum_{k=1}^K \left\| A\tilde{m}_k(n) + B - \frac{\tilde{m}_k(n)}{\|\tilde{m}_k(n)\|} \right\|^2 \quad (3.28)$$

where $n = 1, 2, \dots, N$ denotes the n^{th} iteration. Let A_n and B_n be the resulting matrices from the minimization of (3.28). Every iteration ends with using A_n and B_n to update the estimates of the magnetic field vectors as

$$\tilde{m}_k(n+1) = A_n\tilde{m}_k(n) + B_n, \quad k = 1, 2, \dots, K. \quad (3.29)$$

From (3.29) we can express the magnetic field estimates $\tilde{m}_k(n)$ using the measurement vectors y_k as

$$\tilde{m}_k(n) = \tilde{A}_n y_k + \tilde{B}_n, \quad k = 1, 2, \dots, K \quad (3.30)$$

where \tilde{A}_n and \tilde{B}_n are iteratively defined as

$$\tilde{A}_n = A_n \tilde{A}_{n-1} \text{ and } \tilde{B}_n = A_n \tilde{B}_{n-1} + B_n. \quad (3.31)$$

To determine when the algorithm has reached an acceptable solution¹, we define the following cost

$$J_{stop}(A_n, B_n) = \|B_n\| + \|A_n - I_{3 \times 3}\|. \quad (3.32)$$

The iterations stop when J_{stop} is sufficiently small and the derived matrices \tilde{A}_N and \tilde{B}_N are related to the calibration parameters T and h of the measurement model (3.2) as follows

$$T = \tilde{A}_N^{-1} \text{ and } h = -\tilde{A}_N^{-1} \tilde{B}_N. \quad (3.33)$$

Finally, the estimates $\tilde{m}_k(N)$, $k = 1, 2, \dots, K$, derived at the N^{th} iteration represent the calibrated measurement vectors.

¹The original manuscript does not provide an explicit condition to stop iterations. However it is reasonable to terminate the algorithm when contribution of the updated A_n and B_n to the calibration parameters \tilde{A}_n and \tilde{B}_n is negligible (see (3.31)).

Algorithm 7: Dorveaux et al. [7]

- Step 1: Initialize $\tilde{m}_k(0)$ using (3.27).
 - Step 2: Minimize (3.28) using least squares and derive A_n and B_n .
 - Step 3: Use A_n and B_n to calculate $\tilde{m}_k(n+1)$ from (3.29).
 - Step 4: Calculate \tilde{A}_n and \tilde{B}_n using (3.31).
 - Step 5: Evaluate the cost function $J_{stop}(A_n, B_n)$ from (3.32).
 - Step 6: Repeat steps 2-5 until J_{stop} is sufficiently small.
 - Step 7: Use \tilde{A}_N and \tilde{B}_N to calculate T and h using (3.33).
-

3.6 Vasconcelos et al. [30]

The authors of [30] consider that magnetometers' measurements lie on a ellipsoid manifold following the measurement model (3.2). First, they derive an initial estimate of the calibration parameters T and h by finding the ellipsoid that fits best to the given data. Then, they use the measurement model of (3.2) to formulate a maximum likelihood estimation problem and derive an improved estimate of the calibration parameters T and h .

From (3.2), the magnetic field vector is expressed as $m = T^{-1}(y - h) - T^{-1}\epsilon$. Assuming that the magnitude of the magnetic field is constant during the calibration procedure we can write the following unconstrained optimization problem to derive T and h

$$\underset{T, h}{\text{minimize}} \quad \sum_{k=1}^K \left(\frac{\|T^{-1}(y_k - h)\| - 1}{\sigma_k} \right)^2. \quad (3.34)$$

Here σ_k denotes the standard deviation of the measurement noise in the k^{th} measurement, assuming it is the same for all three axes. Without loss of generality, the magnitude of the magnetic field is assumed to be equal to one. A possible relaxation of this soft assumption is provided by Springmann [28] who addresses the problem of time-varying bias. To solve (3.34), the authors define the following cost function and then minimize it using the Newton's method

$$J(x) \triangleq \sum_{k=1}^K \left(\frac{\|\hat{T}(y_k - h)\| - 1}{\sigma_k} \right)^2 \quad (3.35)$$

where $\hat{T} = T^{-1}$ and

$$x = \left[\text{vec}(\hat{T})^T \quad h^T \right]^T \quad (3.36)$$

The vector x is updated in every Newton's iteration as follows

$$x^{(+)} = x^{(-)} - \left[\nabla^2 J(x) \Big|_{x=x^{(-)}} \right]^{-1} \nabla J(x) \Big|_{x=x^{(-)}} \quad (3.37)$$

where $\nabla J(x)$ is the gradient vector and $\nabla^2 J(x)$ is the Hessian matrix of the cost function. For both $\nabla J(x)$ and $\nabla^2 J(x)$, the authors in [30] provide analytical expressions which are presented in Appendix 3.12.1.

3.6.1 Initial Estimate

Solving (3.34) using the Newton's method requires a good initial estimate of the calibration parameters, \hat{T} and h . Vasconcelos et al. use a previous work on nonlinear estimators for strapdown magnetometers by Foster and Elkaim [9, 11], to derive a good initial estimate by solving the following pseudo-linear least squares estimation problem.

$$Ap = b \quad (3.38)$$

where, by writing each measurement vector as $y_k = [y_k^x \ y_k^y \ y_k^z]^T$, $k = 1, 2, \dots, K$, it is

$$A = \begin{bmatrix} y_1^{x2} & y_1^x y_1^y & y_1^x y_1^z & y_1^{y2} & y_1^y y_1^z & y_1^x & y_1^y & y_1^z & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_K^{x2} & y_K^x y_K^y & y_K^x y_K^z & y_K^{y2} & y_K^y y_K^z & y_K^x & y_K^y & y_K^z & 1 \end{bmatrix} \quad (3.39)$$

and

$$b = [y_1^{z2} \ y_2^{z2} \ \dots \ y_K^{z2}]^T \quad (3.40)$$

The vector p is derived as

$$p = [A \ B \ C \ D \ E \ G \ H \ I \ J]^T = (A^T A)^{-1} A^T b \quad (3.41)$$

The initial estimates of the calibration parameters are derived as

$$\hat{T}(0) = \begin{bmatrix} \frac{1}{\alpha} & 0 & 0 \\ -\frac{1}{\alpha} \tan(\rho) & -\frac{1}{b} \sec(\rho) & 0 \\ \frac{1}{\alpha} (\tan(\rho) \tan(\lambda) \sec(\phi) - \tan(\phi)) & -\frac{1}{b} \sec(\rho) \tan(\lambda) \sec(\phi) & \frac{1}{c} \sec(\lambda) \sec(\phi) \end{bmatrix} \quad (3.42)$$

and

$$h(0) = \frac{1}{2\alpha_1} [\beta_1 \ \beta_2 \ \beta_3]^T \quad (3.43)$$

where

$$\begin{aligned} a &= \frac{1}{2\alpha_1} (-(4D + E^2)\alpha_2)^{1/2} \\ b &= \frac{1}{2\alpha_1} (-(4A + C^2)\alpha_2)^{1/2} \\ c &= \frac{1}{2\alpha_1} ((4DA - B^2)\alpha_2)^{1/2} \\ \tan(\rho) &= -\frac{1}{2\alpha_1} (2B + EC)(\alpha_1)^{-1/2} \\ \tan(\phi) &= (BE - 2CD)(\alpha_1)^{-1/2} \\ \tan(\lambda) &= E(-\alpha_1\alpha_3^{-1})^{1/2} \end{aligned} \quad (3.44)$$

and

$$\begin{aligned}
\beta_1 &= 2BH + BEI - 2CDI - 4DG + ECH - E^2G \\
\beta_2 &= -2AEI + 4AH - BCI - 2BG + C^2H - CEG \\
\beta_3 &= 4DIA - 2DGC + EGB - IB^2 - 2EHA + CBH
\end{aligned} \tag{3.45}$$

The auxiliary variables α_1 , α_2 and α_3 are defined as

$$\begin{aligned}
\alpha_1 &= -B^2 + DC^2 + 4DA + AE^2 - BEC \\
\alpha_2 &= 4AE^2J - E^2G^2 - 4BECJ + 2ECHG + 2BEIG - 4EHAI - 4DICG - C^2H^2 \\
&\quad + 4DAI^2 + 2CBHI - 4DG^2 + 4DC^2J + 4BHG - 4AH^2 - B^2I^2 - 4B^2J + 16DAJ \\
\alpha_3 &= E^4A - CBE^3 + E^2C^2D - 2B^2E^2 + 8DAE^2 - 4DB^2 + 16D^2A
\end{aligned} \tag{3.46}$$

One contribution of Vasconcelos et al., advancing the existing initial step approach suggested in [9], was the derivation of the aforementioned explicit and non-trivial expressions. In addition, Vasconcelos et al. state that their proposed algorithm is applicable even when the magnitude of the magnetic field is not constant during the measurement, similarly to TWOSTEP and Crassidis et al. algorithm [5].

Algorithm 8: Vasconcelos et al. [30]

Initial Estimate

Step 1: Use the sensors' measurements y_k , $k = 1, 2, \dots, K$ and form A and b according to (3.39) and (3.40) respectively.

Step 2: Calculate p using (3.41)

Step 3: Derive the initial estimates $\hat{T}(0)$ and $h(0)$ using (3.42) and (3.43) respectively.

Newton's Method

Step 4: Use the initial estimates $\hat{T}(0)$ and $h(0)$ to initialize x according to (3.36).

Step 5: Update x using (3.37).

Step 6: Evaluate the cost function $J(x)$ of (3.35).

Step 7: Repeat Steps 5-6 until $J(x)$ becomes sufficiently small.

Step 8: Split x into \hat{T} and h and calculate $T = \hat{T}^{-1}$.

3.7 Ali et al. [1]

The authors propose a Particle Swarm Optimization (PSO) [16] - based calibration algorithm that estimates the bias, the scale and nonorthogonality factors. The main advantage of this algorithm is its simplicity of implementation since the optimization is heuristic and does not depend on calculation of gradients, unlike other optimization techniques mentioned in this paper. It can be classified as an Artificial Intelligence (AI) [17] approach.

The authors in [1] use (3.2) and a set of N sensor's measurements to form the following

optimization problem for deriving the calibration parameters T and h

$$\min_{T,h} J \quad (3.47)$$

where J is called the fitness and is defined by

$$J \triangleq \sqrt{\sum_{k=0}^N \left(\|y_k\|^2 - \|m_k\|^2 \right)^2}. \quad (3.48)$$

Function J depends on T and h which are conveniently combined into the single vector $x \in \mathbb{R}^{12}$,

$$x = \begin{bmatrix} h \\ \text{vec}(T^T) \end{bmatrix} \quad (3.49)$$

For a swarm of S particles, the position $x_i \in \mathbb{R}^{12}$ and the velocity $v_i \in \mathbb{R}^{12}$ of the i -th particle can be computed using [16]

$$v_i^k = v_i^{k-1} + c_1 r_{1i}^{k-1} (p_i^{k-1} - x_i^{k-1}) + c_2 r_{2i}^{k-1} (p_g^{k-1} - x_i^{k-1}) \quad (3.50\alpha')$$

$$x_i^k = x_i^{k-1} + v_i^k \quad (3.50\beta')$$

for $i = 1, 2, \dots, S$ where k denotes the new value while $k - 1$ the old value. Also p_i denotes the i^{th} 's particle best position, p_g denotes the swarm's best position, c_1 and c_2 are the acceleration coefficients, w is the inertial weight factor and r_{1i}, r_{2i} are random numbers uniformly distributed within the range $[0, 1]$. Typical values of these quantities are $c_1 = c_2 = 2$, $w = 1$ and the number of particles S is usually between 20 and 65.

Therefore, at each iteration k , each particle's fitness value $J(x_i^k)$ is calculated and quantities p_i and p_g are updated accordingly. The authors suggest three different stop criteria. Specifically, the iterations stop either when the fitness value J of a particle is smaller than a predetermined threshold, or after a maximum number of iterations, or when the change of J becomes insignificant with iterations. Upon termination of the algorithm, parameters T and h (3.2) are extracted from the swarms's optimal solution p_g according to

$$\begin{bmatrix} h \\ \text{vec}(T^T) \end{bmatrix} = p_g. \quad (3.51)$$

Following the general concept of applying AI optimization algorithms, as was introduced in [1], one can also consider using more modern versions of the standard PSO, e.g. [8, 23, 14]. They are typically found as built-in functions in computational suites such as MATLAB [22].

Algorithm 9: Ali et al. (PSO calibration) [1]

-
- Step 1: Initialize x_i, v_i for $i = 1, 2, \dots, S$
and set $p_i = x_i$
- Step 2: Find $j = \{i | i = 1, 2, \dots, S \text{ and } J(p_i) \leftarrow \min\}$
Particle i best: $J_{min}^i \leftarrow J(p_i)$
Global best: $p_g \leftarrow p_j$ and $J_{min} \leftarrow J(p_j)$
- Step 3: **for each** particle i do
 Update x_i, v_i (3.50)
 Calculate $J(x_i)$ (3.47)
 if $J(x_i) < J_{min}^i$
 $J_{min}^i \leftarrow J(x_i)$ and $p_i \leftarrow x_i$
 if $J(x_i) < J_{min}$
 $J_{min} \leftarrow J(x_i)$ and $p_g \leftarrow x_i$
- Step 4: Repeat Step 3 until an exit condition is met
- Step 5: Extract T and h from p_g (3.51)
-

3.8 Wu and Shi [31]

The authors of [31], formulate the calibration of a three-axis magnetometer as a maximum likelihood estimation problem which is solved using the Gauss-Newton method.

Starting from the measurement model of (3.2), Wu and Shi observed that by considering the QR decomposition $T^{-1} = QR$, where $Q \in O(3)$ and $R \in U(3)$, (3.2) is written as

$$y = R^{-1}Q^T m + h + \varepsilon \quad (3.52)$$

Defining $\hat{m} \triangleq Q^T m$, we observe that $\|\hat{m}\| = \|m\|$ since $Q \in O(3)$. Also setting $\hat{T} \triangleq R^{-1}$ we have that

$$y = \hat{T}\hat{m} + h + \varepsilon \quad (3.53)$$

Using the above transformation, the authors reduce the unknown model parameter variables from 12 (9 for T and 3 for h) to 9 (6 for R since R is upper triangular and 3 for h). Note that using (3.53), the calibration procedure now aims at finding the calibration parameters \hat{T} and h while the magnetic field vector \hat{m} is also unknown.

Using a set of K measurements and (3.53), the authors formulate the following maximum likelihood estimation problem

$$\begin{aligned} & \underset{\hat{T}, h, \hat{m}_k}{\text{minimize}} && \sum_{k=1}^K \|y_k - \hat{T}\hat{m}_k - h\|^2 \\ & \text{subject to} && \|\hat{m}_k\| = 1, \quad k = 1, 2, \dots, K. \end{aligned} \quad (3.54)$$

Without loss of generality, the authors, constrained the magnitude of the magnetic field to be

equal to one. Based on (3.54), the following Lagrange function is formulated

$$J(x) = \sum_{k=1}^K \left[\|y_k - \hat{T}\hat{m}_k - h\|^2 + \lambda_k (\|\hat{m}_k\|^2 - 1) \right] \quad (3.55)$$

where

$$x = \left[\text{vec}(\hat{T})^T, h^T, \hat{m}_1^T, \hat{m}_2^T, \dots, \hat{m}_K^T, \lambda_1, \lambda_2, \dots, \lambda_K \right]^T \quad (3.56)$$

and $\lambda_k, k = 1, 2, \dots, K$ are positive Lagrange coefficients for the unit norm constrain. Note that since \hat{T} is an upper triangular matrix, the lower triangular elements of \hat{T} are excluded from x . The minimization of (3.55) and the estimation of x are done using the Gauss-Newton method as follows

$$x^{(+)} = x^{(-)} - \left[\nabla^2 J(x) \Big|_{x=x^{(-)}} \right]^{-1} \left(\nabla J(x) \Big|_{x=x^{(-)}} \right) \quad (3.57)$$

where $\nabla J(x)$ is the Jacobian vector and $\nabla^2 J(x)$ is the Hessian matrix of the Lagrange function. For both $\nabla J(x)$ and $\nabla^2 J(x)$, the authors provide analytical expressions which are presented in Appendix 3.12.2.

3.8.1 Initial Estimate

Solving (3.54) using the Gauss-Newton method requires a good initial estimate of the unknowns. To find one, the authors of [31] use the unit magnitude constrain and the equation $1 = \|R(y_k - h)\|^2$ which after some manipulation, is written as

$$\begin{bmatrix} y_k^T \otimes y_k^T & y_k^T & 1 \end{bmatrix} \begin{bmatrix} \text{vec}(A) \\ b \\ c \end{bmatrix} \triangleq Y_k z = 0, \quad k = 1, 2, \dots, K \quad (3.58)$$

where $A = R^T R$, $b = -2R^T R h$ and $c = h^T R^T R h$. Defining $Y = \begin{bmatrix} Y_1^T & Y_2^T & \dots & Y_K^T \end{bmatrix}^T$, from (3.58) it is

$$Y z = 0 \quad (3.59)$$

The authors, solve (3.59) in a least squares sense and denote the solution $z_e = \left[\text{vec}(A_e)^T \quad b_e^T \quad c_e \right]^T = \min \|Y z\|^2$. They derive z_e as the eigenvector of $Y^T Y$ corresponding to its minimum (or zero) eigenvalue. Using z_e , the vector z is derived as $z = \alpha z_e$, where $\alpha = 4 / (b_e^T A_e^{-1} b_e - 4c_e)$. Extracting $\text{vec}(A)$, b and c from z , the initial estimates of the unknowns, $\hat{T}(0), h(0), \hat{m}_k(0)$ and $\lambda_k(0)$ are defined as follows:

$$\begin{aligned} \hat{T}(0) &= R^{-1} = \text{chol}(A) \\ h(0) &= -A^{-1} b / 2 \\ \hat{m}_k(0) &= \hat{T}(0)^{-1} (y_k - h), \quad k = 1, 2, \dots, K \\ \lambda_k(0) &= 0, \quad k = 1, 2, \dots, K \end{aligned} \quad (3.60)$$

where $\text{chol}(\cdot)$ is the Cholesky factorization.

Finally, an alternative version of Wu's and Shi's algorithm is proposed by Cao et al. in [4], where a different method for the initial estimate is presented, and the second step is identical.

Algorithm 10: Wu and Shi [31]

Initial Estimate

Step 1: Calculate Y_k , $k = 1, 2, \dots, K$ from (3.58) and form the matrix

$$Y = [Y_1^T \ Y_2^T \ \dots \ Y_K^T]^T.$$

Step 2: Find the eigenvector of $Y^T Y$ corresponding to its minimum (or zero) eigenvalue and denote it as $z_e = [\text{vec}(A_e)^T \ b_e^T \ c_e]^T$.

Step 3: Calculate $z = \alpha z_e$ where $\alpha = 4 / (b_e^T A_e^{-1} b_e - 4c_e)$.

Step 4: Extract $\text{vec}(A)$, b and c from z .

Step 5: Calculate an initial estimate of the unknowns using (3.60).

Gauss-Newton Method

Step 6: Use the initial estimates to initialize the vector x of (3.56)

Step 7: Update x using (3.57).

Step 8: Evaluate the cost $J(x)$ of (3.55).

Step 9: Repeat steps 7-8 until $J(x)$ becomes sufficiently small.

3.9 Papafotis and Sotiriadis (MAG.I.C.AL.) [25]

The authors in [25] use (3.2) and a set of K sensor's measurements to form the following optimization problem for deriving the calibration parameters T and h

$$\begin{aligned} & \underset{T, h, m_k}{\text{minimize}} && \sum_{k=1}^K \|y_k - T m_k - h\|^2 \\ & \text{subject to} && \|m_k\| = 1, \quad k = 1, 2, \dots, K \end{aligned} \quad (3.61)$$

where, without loss of generality, the magnitude of the magnetic field is constrained to be equal to one. In order to solve (3.61) they propose an iterative algorithm, based on the solution of a linear least-squares problem.

The algorithm begins by initializing the magnetic field vectors, m_k , as

$$m_k = \frac{y_k}{\|y_k\|}, \quad k = 1, 2, \dots, K \quad (3.62)$$

and rewriting (3.2) in a matrix form as follows:

$$Y = LG + E \quad (3.63)$$

where

$$Y = \begin{bmatrix} y_1 & y_2 & \dots & y_K \end{bmatrix} \quad (3.64\alpha')$$

$$L = \begin{bmatrix} T & h \end{bmatrix} \quad (3.64\beta')$$

$$G = \begin{bmatrix} m_1 & m_2 & \dots & m_K \\ 1 & 1 & \dots & 1 \end{bmatrix} \quad (3.64\gamma')$$

$$E = \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & \dots & \varepsilon_K \end{bmatrix} \quad (3.64\delta')$$

In every iteration, (3.63) is solved for L using the least squares method, minimizing the total squared error $\|E^T E\|_F^2$

$$L = YG^T(GG^T)^{-1} \quad (3.65)$$

From the calculated L , an updated set of calibration parameters T and h is extracted from (3.64 β'). Using them, the magnetic field vector is updated as

$$m_k = \frac{\tilde{m}_k}{\|\tilde{m}_k\|}, \quad k = 1, 2, \dots, K \quad (3.66)$$

where

$$\tilde{m}_k = T^{-1}(y_k - h), \quad k = 1, 2, \dots, K \quad (3.67)$$

Every iteration ends by updating the matrix G using the updated vectors m_k , $k = 1, 2, \dots, K$. Iterations stop when a small value of the following cost function is achieved

$$J(T, h) = \sum_{k=1}^K (\|\tilde{m}_k\|^2 - 1)^2 \quad (3.68)$$

Algorithm 11: Papafotis and Sotiriadis (MAG.I.C.AL.) [25]

- Step 1: Initialize m_k using (3.62).
 - Step 2: Calculate L using (3.65).
 - Step 3: Extract T and h from L using (3.64 β').
 - Step 4: Update m_k using (3.66) and (3.67) and use it to update G .
 - Step 5: Evaluate the cost-plus-penalty function J from (3.68).
 - Step 6: Repeat steps 2-5 until $J(T, h)$ is sufficiently small.
-

3.10 Algorithms' Evaluation and Comparison

In this Section, the performance of the presented algorithms is evaluated in terms of accuracy, robustness and execution speed. Firstly, we evaluate the performance of the seven algorithms using multiple sets of synthetic data where the calibration parameters T and h as well as the measurement noise characteristics are predefined and known. By doing so, we are able to accurately determine the algorithms' accuracy and robustness. Then multiple datasets of two different low-cost magnetic field sensors are used to verify the algorithms' performance under real-world conditions.

3.10.1 Synthetic Data Generation

We designed a procedure to generate synthetic data effectively, in order to examine each of the aforementioned algorithm's performance across a range of noise variance and measurement sample size. The authors of TWOSTEP [3] propose a typical scenario of assuming the magnetic vector spinning with a constant angular velocity. On the other hand, Wu and Shi [31] suggest a specific sequence of 3D rotations using Euler Angles, applied on a constant known magnetic vector m . In the same page, Papafotis and Sotiriadis [25] recommend a sequence of 12 approximate orientations. Another alternative is to make use of a set of random, yet normalized, vector fields, which however demands a reasonable amount of samples.

Because none of the described algorithms guarantees that it will function properly under an arbitrary data set, we propose an efficient method to span $SO(3)$, following [26], so as to provide the algorithms with substantial, non-redundant information and to compare them fairly. After extensive simulation, it was observed that the recommended method was very effective in spanning the 3D rotation space.

Our method's effectiveness lies on distributing the points on the sphere $\|m\| = 1$, more evenly by using the canonical Fibonacci Lattice mapping [26, 12]. Generating a Fibonacci sphere is an extremely fast and effective approximate method to evenly distribute points on a sphere.

This way $SO(3)$ is sufficiently represented even with only a small dataset. An algorithm for generating K vectors distributed on a Fibonacci sphere is presented in detail in Algorithm 12.

Considering K vectors, m_k , $k = 1, 2, \dots, K$ distributed on a Fibonacci sphere, we continue with generating matrix T and vector h , required to calculate the corresponding measurement vectors y_k , m_k , $k = 1, 2, \dots, K$ according to (3.2). Ideally, matrix T would be the 3×3 identity matrix while the bias vector h would be the 3×1 vector of zeros. A realistic model for T and h , accounting for the sensor's non-idealities, is derived by using the concept of additive perturbation

$$T = \alpha I_3 + E \quad (3.69\alpha')$$

$$h = e \quad (3.69\beta')$$

where α accounts for gross scaling errors, E is a 3×3 perturbation matrix with random, typically small, coefficients and e is 3×1 perturbation bias vector with random coefficients. Finally, a sequence of white noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ is added to the measurements and the measurement vectors y_k , m_k , $k = 1, 2, \dots, K$ are derived according to (3.2)

$$y = Tm + h + \varepsilon \quad (3.70)$$

Algorithm 12: Generation of Synthetic Data

Step 1: Initialize the number of measurements K and the radius of sphere r

Step 2: Calculate Golden Ratio: $\varphi = \frac{1+\sqrt{5}}{2}$

Step 3: **for each** $k = 1, 2, \dots, K$ **do**:

$$\theta = \frac{2\pi k}{\varphi}$$

$$\phi = \arccos\left(1 - \frac{2(k+0.5)}{K}\right)$$

$$m_k = [m_x, m_y, m_z] = [r \cos \theta \sin \phi, r \sin \theta \sin \phi, r \cos \phi]$$

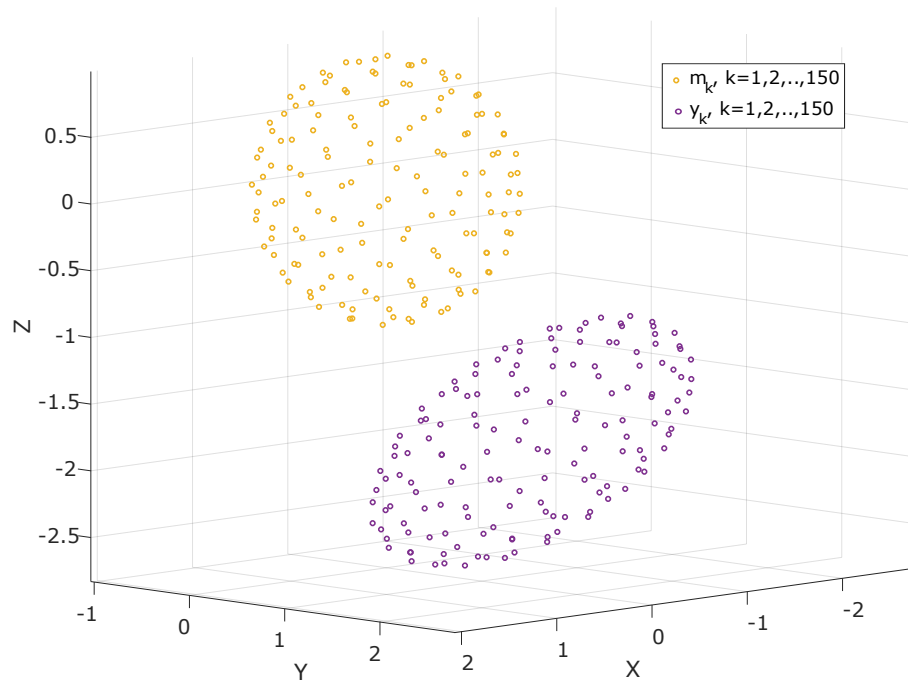
Step 4: Pick the scaling parameter, α , the perturbation matrix, E and the perturbation vector, e .

Step 5: Calculate T and h according to (3.69).

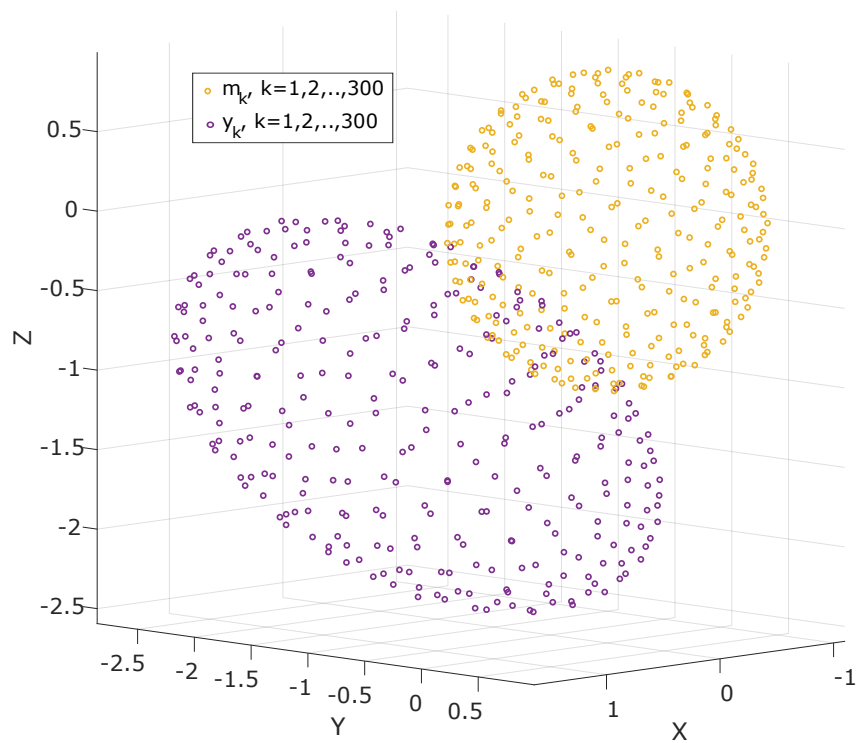
Step 6: Generate a sequence of white noise: $\varepsilon \sim \mathcal{N}(0, \sigma^2)$

Step 7: Calculate the measurement vectors: $y_k = Tm_k + h + \varepsilon_k$ (3.2)

The two datasets generated using Algorithm 12 are presented in Figure 3.1. Note that for visualization purposes, the scaling parameter, α , the perturbation matrix, E , and the perturbation vector, e , used to create each dataset were set to a rather large value.



(α') Synthetic dataset generated using Algorithm 12 for $K = 150$.



(β') Synthetic dataset generated using Algorithm 12 for $K = 300$.

Figure 3.1: Two synthetic datasets generated using Algorithm 12 for $K = 150$ (a) and $K = 300$ (b) respectively.

Experiment Setup and Evaluation Criteria

To evaluate the algorithms' performance, we used synthetic data, generated by Algorithm 12, and we executed a great number Monte Carlo simulations. Each simulation consisted of 250 runs of each algorithm while in each run, the same dataset was used as input in all algorithms. An uncertainty was introduced in the generation of each dataset by considering a statistical distribution for the elements, E_{ij} , of the perturbation matrix, E , and the elements, e_i , of the perturbation vector e (see (3.69)). Specifically, for the Monte Carlo simulations we assumed

$$\alpha \sim \mathcal{U}[0.8, 1.2] \quad (3.71\alpha')$$

$$E_{ij} \sim \mathcal{U}[-\beta, \beta] \quad (3.71\beta')$$

$$e_i \sim \mathcal{U}[-\gamma, \gamma] \quad (3.71\gamma')$$

where β and γ are scalars, the effect of which was tested using multiple Monte Carlo simulations. Note that we considered the scaling factor, α , to be close to the ideal value of $\alpha = 1$. That may not be the case when real-world measurements are used, however, it is trivial, and common, to properly scale the measurements before the calibration procedure and remove gross scaling errors. In this way, the algorithms are not burdened, searching for a scaling relationship which can be easily provided by simple data preprocessing.

A challenging point while setting up the experiments was to determine the number of samples of each dataset and the value of the sensor's noise variance, σ^2 . We considered a dataset of 300 measurements as a solid choice for a simulation environment based on [31, 1] while we experimentally confirmed that bigger datasets do not improve the performance of any algorithm. We also examined the performance of the presented algorithms when smaller datasets, consisted of 150 and 50 measurements, are used. As far as the noise variance, σ^2 , is concerned, we considered a nominal value of $\sigma = 0.005$, following [30] and [31] while we also simulated the cases of more noisy ($\sigma = 0.05$) and less noisy ($\sigma = 0.0005$) sensors.

The evaluation of the algorithm for each Monte Carlo simulation was done in terms of accuracy, execution speed and robustness. We used the execution speed of each algorithm as a metric of computational efficiency and is defined as the inverse of the mean execution time. As a metric of robustness we considered the percentage of datasets for which each algorithm successfully derived a meaningful solution.

The definition of an accuracy metric is a little more involved. Each algorithm was developed to take as inputs the measurement vectors y_k , $k = 1, 2, \dots, K$ and output the calibration parameters T and h . Comparing the output bias vector h with the true one, h_{true} , which was used in the data generation procedure, was done by defining the following cost

$$J_h = \|h_{true} - h\| \quad (3.72)$$

The calibration matrix T on the other hand is derived under a rotational uncertainty and comparing it with the true one, T_{true} , is a more challenging task.

Consider the measurement model of (3.2). Noting that the true magnetic field vector in (3.2) is also unknown, and derived by the calibration algorithm, we can write:

$$y = T_{true}RR^T m + h_{true} \quad (3.73)$$

where R is an orthogonal matrix in the $O(3)$ group. Thus, taking into account the rotational invariance of the Euclidean norm which implies that $\|R^T m\| = \|m\|$, a calibration algorithm may output any matrix T of the form $T = T_{true}R$. Thus a proper cost function to compare T and T_{true} is the following

$$J_T = \|T - T_{true}R\|_F \quad (3.74)$$

where, the matrix R is defined as the solution of the following minimization problem

$$R = \underset{\Omega \in O(3)}{\operatorname{argmin}} \|T - T_{true}\Omega\|_F \quad (3.75)$$

The solution of (3.75) is given by the Orthogonal Procrustes problem [27], and it is

$$R = UV^T \quad (3.76)$$

where the matrices U and V are derived from the Singular Value Decomposition (SVD) of the matrix $T_{true}^T T$, i.e. $T_{true}^T T = U\Sigma V^T$, where $U, V \in O(3)$ and Σ is a diagonal matrix.

Using (3.72) and (3.74) we define the following cost function as a metric of accuracy.

$$J = \|h_{true} - h\| + \|T - T_{true}R\|_F \quad (3.77)$$

Based on the above and given the results of a Monte Carlo simulation consisted of N executions of each algorithm, we define the following metrics of performance:

- **Accuracy** is defined as the mean value of the cost J , defined in (3.77), across all N executions with meaningful output.
- **Mean Execution Time** is defined as the mean value of the execution time of an algorithm.
- **Robustness** is defined as the percentage of datasets for which each algorithm successfully derived a meaningful solution.

The robustness criterion can be seen as the frequency in which an algorithm provides a better solution (T, h) in the sense of the cost function (3.77), than the trivial solution $(I_{3 \times 3}, 0_{3 \times 1})$ which assumes no bias and non multiplicative errors. Given the cost J_o that corresponds to the trivial solution,

$$J_o = \|h_{true} - 0_{3 \times 1}\| + \|I_{3 \times 3} - T_{true}R\|_F \quad (3.78)$$

an execution of an algorithm is considered as successful with meaningful output when

$$J < \delta J_o \quad (3.79)$$

where $\delta \in (0, 1)$ is a robustness parameter. If δ is close to 1, it means that only little improvement with respect to J_o is sufficient. As δ gets smaller, better solutions are required. Thus, this parameter can be tuned with respect to the test's objective and the application's specifications. Given N runs for an algorithm, its robustness is denoted by $RB(\%)$ and is defined as

$$RB(\%) = \frac{1}{N} \sum_{i=1}^N \mathbb{U}(J_i < \delta J_{oi}) \cdot 100. \quad (3.80)$$

Here J_i and J_{oi} are the values of J (3.77) and J_o (3.78), respectively, corresponding to the i^{th} run of the algorithm and \mathbb{U} is a boolean function, which is one if its argument is true and zero otherwise. Let M denote the number of executions meaningful outputs.

Now, the accuracy metric is only applied on the M meaningful outputs according to the robustness test (3.79), since otherwise the comparison would be unfair for the least stable algorithms. The accuracy of an algorithm over a dataset is denoted by ρ and it is defined as

$$\rho = \frac{1}{M} \sum_{i=1}^M \mathbb{U}(J_i < \delta J_{oi}) J_i \quad (3.81)$$

which is the mean accuracy metric value over the M executions with meaningful outputs.

Similarly, the time-efficiency metric (i.e. Mean Execution Time) is only applied on the M executions with meaningful outputs according to the robustness test (3.79). Again, this is because otherwise the comparison would be unfair for the least stable algorithms. The Mean Execution Time of an algorithm over a dataset, is denoted by τ and is defined as

$$\tau = \frac{1}{M} \sum_{i=1}^M \mathbb{U}(J_i < \delta J_{oi}) t_i \quad (3.82)$$

where t_i is the time needed for the i run to be completed. The execution speed of an algorithm is defined as $1/\tau$.

Baseline Evaluation

To derive a baseline evaluation of the presented algorithms, we run a Monte-Carlo simulation considering typical values for the sensor's error and noise parameters. In this simulation we neglected the effect of hard-iron and soft-iron distortions which are in some cases the dominant terms of the overall error as well as extreme cases of large manufacturing imperfections. More specifically, 250 different datasets consisting of 300 measurements each, were generated following Algorithm 12 and considering the following distributions of the model disturbances and the

Algorithm	Accuracy ($1/\rho$)	Robustness ($RB\%$)	Execution Speed ($1/\tau$)
TWOSTEP [2]	$35.3 \cdot 10^0$	91.6%	455 s^{-1}
Crassidis et al. [5]	$3.31 \cdot 10^3$	100%	47.6 s^{-1}
Dorveaux et al. [7]	$2.26 \cdot 10^5$	100%	12.8 s^{-1}
Vasconcelos et al. [30]	$2.28 \cdot 10^5$	99.6%	0.089 s^{-1}
Ali et al. [1]	$2.27 \cdot 10^5$	98.8%	0.10 s^{-1}
Wu and Shi [31]	$2.32 \cdot 10^5$	87.2%	0.24 s^{-1}
MAG.I.C.AL [25]	$2.28 \cdot 10^5$	100%	29.4 s^{-1}

Table 3.2: Baseline Evaluation of the presented algorithms.

measurement noise

$$\begin{aligned}
\alpha &\sim \mathcal{U}[0.8, 1.2] \\
E_{ij} &\sim \mathcal{U}[-0.05, 0.05] \\
e_i &\sim \mathcal{U}[-0.05, 0.05] \\
\sigma &= 0.005
\end{aligned} \tag{3.83}$$

The distributions' ranges in (3.83) are based on our literature review. The selection $\beta = \gamma = 0.05$ corresponds to the typical case of approximately 5% distortion for T and bias h . The measurement's noise standard deviation is set to a typical value of $\sigma = 0.005$ [31, 30].

The performance of the seven algorithms is presented in Table 3.2.

The effect of the offset perturbation parameter, γ

Under extreme manufacturing imperfections or the effect of hard-iron distortion, the magnitude of the offset vector, h , can be much larger than that in the typical case. In this Section we examine how larger values of $\|h\|$ affect the performance of the presented algorithms. To do so, we run six Monte-Carlo simulations, each one comprised of 250 different datasets generated by following Algorithm 12. The offset vector perturbation parameter e_i is simulated with gradually increasing magnitude by expanding the selection horizon $\mathcal{U}[-\gamma, \gamma]$. Afterwards, its corresponding impact on each algorithm's robustness and accuracy is investigated. The distributions of the model disturbances and measurement noise are:

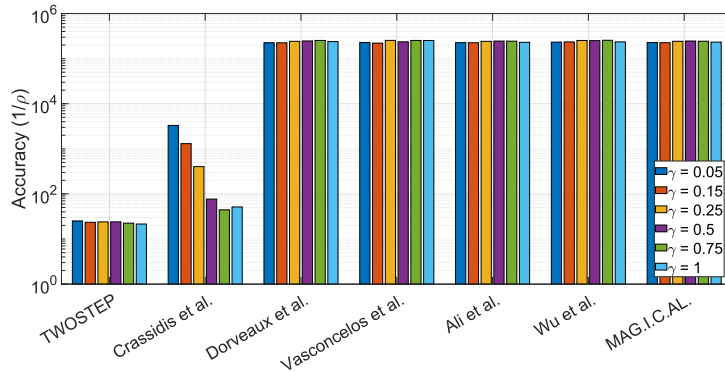
$$\begin{aligned}
\alpha &\sim \mathcal{U}[0.8, 1.2] \\
E_{ij} &\sim \mathcal{U}[0.05, 0.05] \\
e_i &\sim \mathcal{U}[-\gamma_l, \gamma_l] \\
\sigma &= 0.005
\end{aligned}$$

for various γ

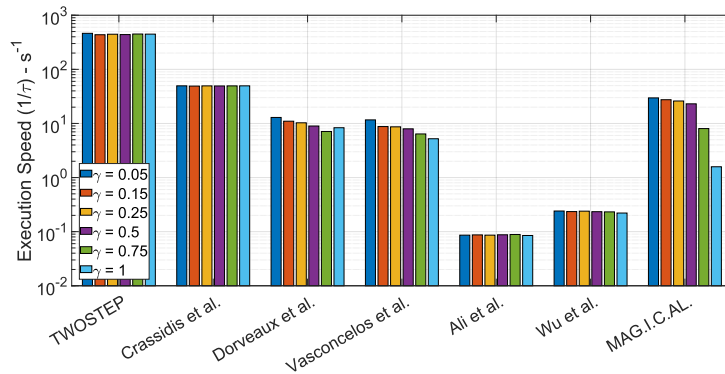
$$\gamma = \{0.05, 0.15, 0.25, 0.5, 0.75, 1\}$$

where $l = 1, 2, \dots, 6$ is the index of Monte Carlo simulation. The extreme case of $\gamma = 1$ addresses the possibility of bias being clearly comparable and even indistinguishable to the true magnetic

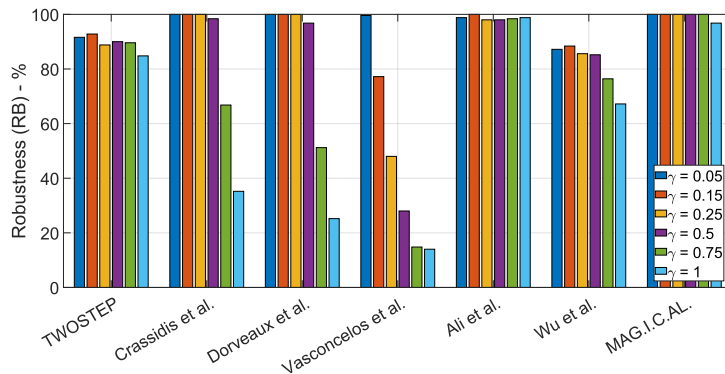
vector. Therefore, as γ increases, the algorithms were driven to their limits and their functionality range was identified. All the other parameters were nominal, to ensure a fair comparison. The results of the six Monte-Carlo simulations are presented in Figure 3.2.



(α) Accuracy ($1/\rho$) of the presented algorithms for different values of γ .



(β) Execution Speed ($1/\tau$) of the presented algorithms for different values of γ .



(γ) Robustness ($RB(\%)$) of the presented algorithms for different values of γ .

Figure 3.2: Performance characteristics of the presented algorithms for different values of γ .

The MAG.I.C.AL and the PSO methods are the most robust ones since they function almost always, even for large values of bias, while TWOSTEP and Wu and Shi's algorithms are a little less stable. In addition, Dorveaux et al. algorithm and EKF seem to be reliable for small to moderate values of bias. All algorithms, except TWOSTEP and EKF are extremely precise when they function properly. No changes in execution speed are noticed, with the exception of MAG.I.C.AL which probably requires more iterations as the bias increases.

The effect of the calibration matrix perturbation parameter, β

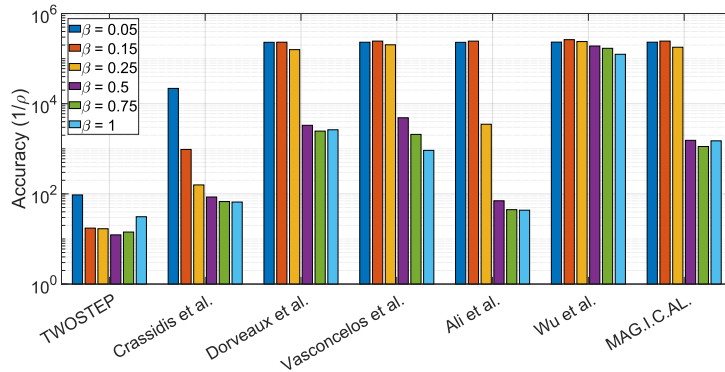
Similar to the case of the offset vector, h , under extreme manufacturing imperfections or the effect of soft-distortion, matrix T , can also diverge significantly from the typical case of the identity matrix. In this Section we examine how larger values of perturbation E affect the performance of the presented algorithms. To do so, we run six Monte-Carlo simulations, each one based on 250 different datasets generated by following Algorithm 12. The perturbation elements E_{ij} were simulated with gradually increasing magnitude by expanding the distribution range $\mathcal{U}[-\beta, \beta]$. Afterwards, its corresponding impact on each algorithm's robustness and accuracy is investigated. The distributions of the model disturbances and measurement noise are:

$$\begin{aligned}\alpha &\sim \mathcal{U}[0.8, 1.2] \\ E_{ij} &\sim \mathcal{U}[-\beta_l, \beta_l] \\ e_i &\sim \mathcal{U}[-0.05, 0.05] \\ \sigma &= 0.005\end{aligned}$$

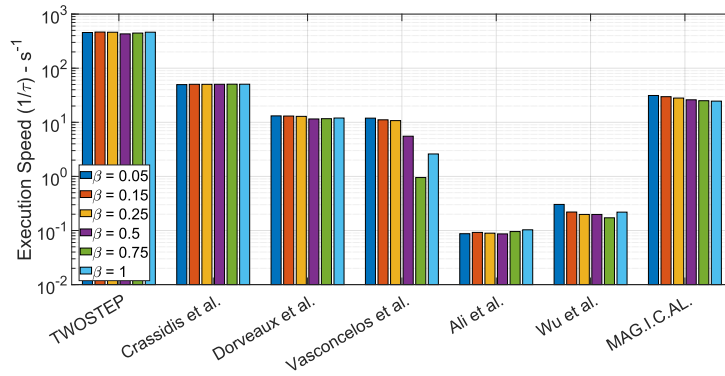
for various β

$$\beta = \{0.05, 0.15, 0.25, 0.5, 0.75, 1\}$$

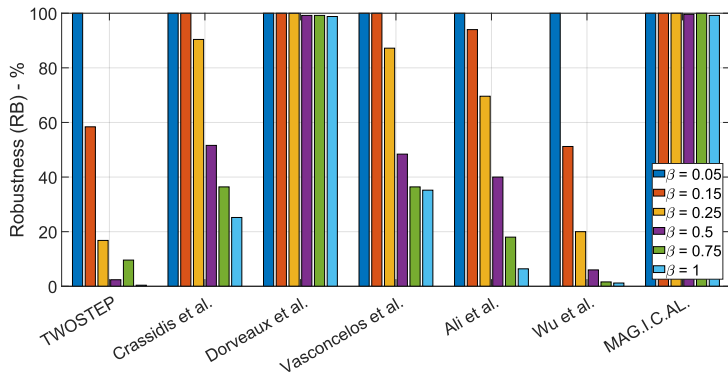
where $l = 1, 2, \dots, 6$ is the index of Monte Carlo simulation. As β increases, the algorithms were driven to their limits and their functionality range was identified. All the other parameters were nominal, to ensure a fair comparison. The results of the six Monte-Carlo simulations are presented in Figure 3.3.



(α) Accuracy ($1/\rho$) of the presented algorithms for different values of β .



(β) Execution Speed ($1/\tau$) of the presented algorithms for different values of β .



(γ) Robustness ($RB(\%)$) of the presented algorithms for different values of β .

Figure 3.3: Performance characteristics of the presented algorithms for different values of β .

The MAG.I.C.AL algorithm and the algorithm of Dorveaux et al. appear to be the most robust and effective, with similar accuracy. The algorithm of Vasconcelos et al., the PSO algorithm and the EKF algorithm succeed only for small to moderate non-orthogonality errors. Vasconcelos et al. achieves accuracy comparable to that of MAG.I.C.AL. The rest of the algorithms tend to fail frequently as these errors increase. What is surprising is that Wu and Shi's algorithm

provides the most accurate solutions for all β values, but with very low robustness. To conclude, most algorithms handle bias distortion better than non-orthogonality errors.

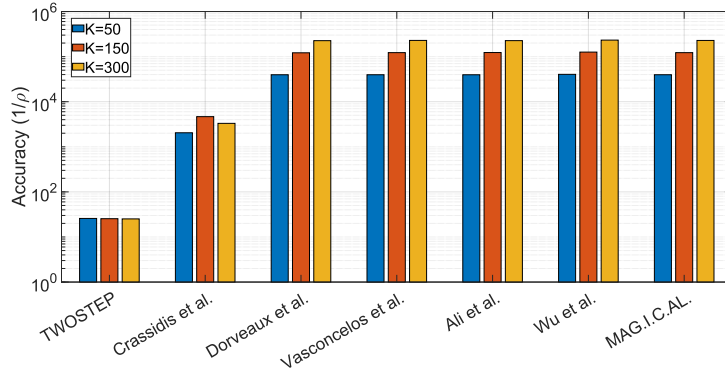
The effect of dataset size, K

In this section, we examine how the dataset size, K , affects the algorithms' performance. In general, the diversity of the measurement directions is more crucial than the quantity of them. E.g. a dataset of 50 measurements with directions distributed near uniformly on the unit sphere is significantly more suitable for the algorithms than one with thousands of measurements all having approximately the same direction.

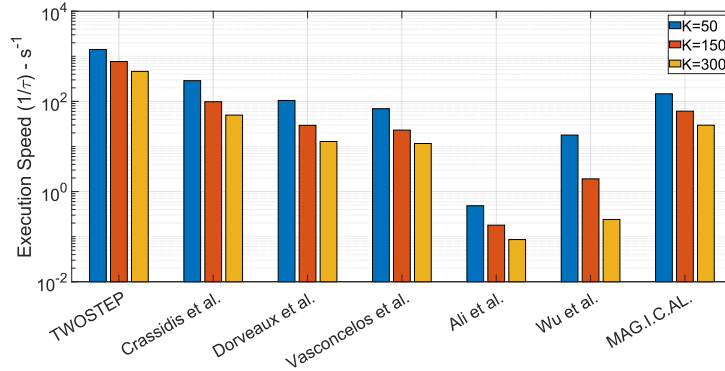
According to existing literature [31, 1, 25], an order of 300 measurements with directions sufficiently covering the unit sphere form an acceptable dataset for the calibration. Here we use datasets with 50, 150 and 300 measurements to test the algorithms' limits. To do so, we run three Monte-Carlo simulations, based on 250 different datasets generated by Algorithm 12. The distributions of the model disturbances and measurement noise are:

$$\begin{aligned}\alpha &\sim \mathcal{U}[0.8, 1.2] \\ E_{ij} &\sim \mathcal{U}[-0.05, 0.05] \\ e_i &\sim \mathcal{U}[-0.05, 0.05] \\ \sigma &= 0.005\end{aligned}$$

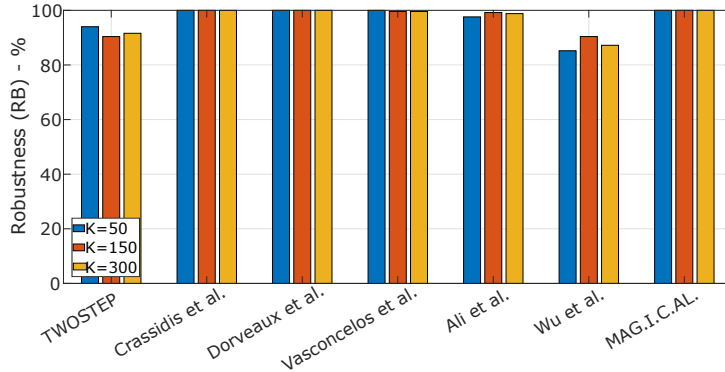
The dataset size K varied whereas the distributions' ranges were fixed to nominal to ensure a fair comparison. The results of the three Monte-Carlo simulations are presented in Figure 3.4.



(α) Accuracy ($1/\rho$) of the presented algorithms for datasets of different size K .



(β) Execution Speed ($1/\tau$) of the presented algorithms for datasets of different size K .



(γ) Robustness ($RB(\%)$) of the presented algorithms for datasets of different size K .

Figure 3.4: Performance characteristics of the presented algorithms for different values of K .

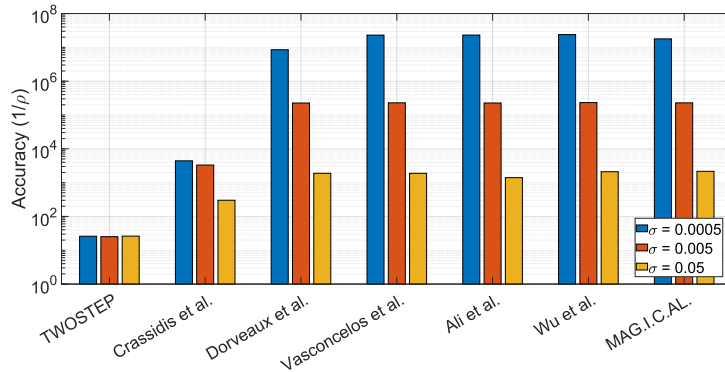
In general, the dataset size, K , does not seem to be important in terms of robustness. Accuracy is surprisingly high even with only 50 measurements, which is probably an outcome of the well distributed measurement directions using the Fibonacci lattice. Furthermore, the algorithms execution time appeared to be linear with K .

The effect of the noise variance, σ^2

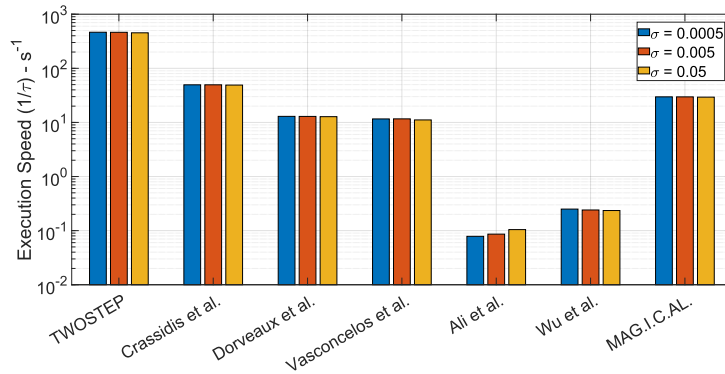
In this section, we examine the influence of measurement's noise variance σ on algorithms' robustness and accuracy. The assumption of pure white Gaussian noise in the measurement's model was done. We considered a nominal value of $\sigma = 0.005$, following [30] and [31] while we also simulated the cases of more noisy ($\sigma = 0.05$) and less noisy ($\sigma = 0.0005$) sensors. With these choices, we represented algorithms' capabilities under 3 different orders in the magnitude of the error in the measurement. To do so, we run three Monte-Carlo simulations, each one based on 250 different datasets generated by following Algorithm 12. The distributions of the model disturbances and measurement noise are:

$$\begin{aligned}\alpha &\sim \mathcal{U}[0.8, 1.2] \\ E_{ij} &\sim \mathcal{U}[-0.05, 0.05] \\ e_i &\sim \mathcal{U}[-0.05, 0.05] \\ \sigma &= \{0.0005, 0.005, 0.05\}\end{aligned}$$

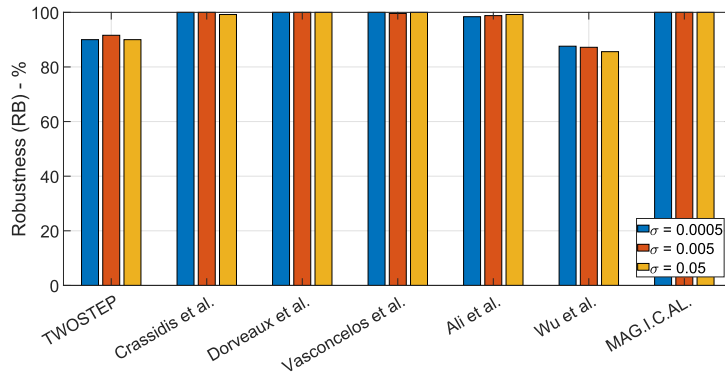
Finally, all parameters except σ were set to their default ones, to ensure a fair comparison. The results of the three Monte-Carlo simulations are presented in Figure 3.5.



(α') Accuracy ($1/\rho$) of the presented algorithms for different values of the noise variance, σ^2 .



(β') Execution Speed ($1/\tau$) of the presented algorithms for different values of the noise variance, σ^2 .



(γ') Robustness ($RB(\%)$) of the presented algorithms for different values of the noise variance, σ^2 .

Figure 3.5: Performance characteristics of the presented algorithms for different values of the noise variance, σ^2 .

All algorithms appear to be immune to the change of measurement's output variance σ . What is worth mentioning is that an increase of one order in variance resulted to a decrease of one order in accuracy for most algorithms (i.e. MAG.I.C.AL, Ali et al., Vasconcelos et al., Dorveaux et al., Wu and Shi). Low accuracy algorithms, like TWOSTEP and EKF showed a

	BNO055	LSM9DS1TR
Measurement Range	± 13 Gauss	± 4 Gauss
Sampling Rate	30Hz	80Hz
Measurement Resolution	16 bits	16 bits

Table 3.3: Operation parameters of the two magnetic field sensors.

constant performance.

3.10.2 Algorithms' Evaluation Using Real Data

In this section, the aforementioned algorithms are tested using real data. Multiple datasets captured by low-cost magnetic field sensors were used to verify the algorithms' performance under real-world conditions. In this case parameters T_{true} and h_{true} are not known in advance. Therefore, the accuracy metric (3.77) cannot be used. Since, the measurements took place in a specific location, a constant magnitude of magnetic vector, $\|m\| = 1$ was considered. As a result, a proper cost function to evaluate algorithm's effectiveness is the following

$$J_r = \frac{1}{K} \sum_{i=1}^K (\|m_k\|^2 - 1)^2 \quad (3.84)$$

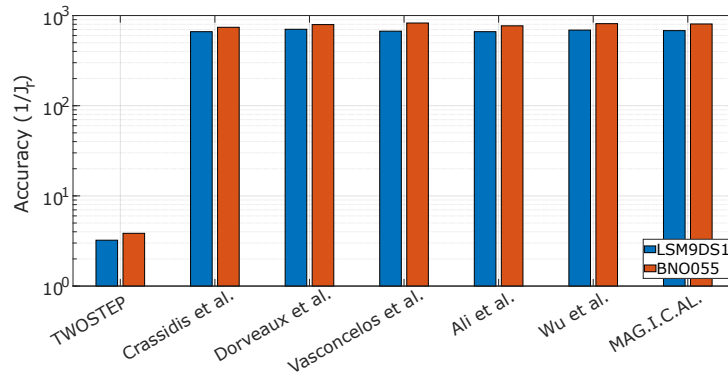
where K is the number of measurements and $k = 1, 2, \dots, K$ is the measurement's index. The estimated magnetic field vector m_k for each k is given by

$$m_k = T^{-1}(y_k - h) \quad (3.85)$$

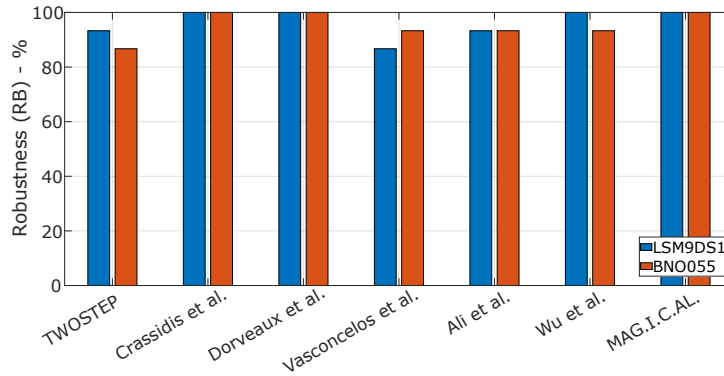
where T and h are the outputs of a calibration algorithm. Such a cost function is described by Wu and Shi (3.55), as well as by Papafotis and Sotiriadis (3.68).

To evaluate the performance of the presented algorithms, we used two off-the-shelf, low-cost magnetic field sensors, which are typically found in commercial electronic devices such as smartphones, activity trackers, etc. More specifically, we captured a total of 30 datasets using the LSM9DS1 by STMicroelectronics and the BNO055 by Bosch Sensortec. The operation parameters of the two sensors during the experiment are presented in Table 3.3. During the experiment, two sensors were fixed on the same rigid platform which was rotated by hand in several orientations.

In Figure 3.6 α' , the mean value of the cost function (3.84) across all the recorded datasets for every algorithm is presented as a metric of accuracy. The robustness of each algorithm, as defined in (3.80) is presented in Figure 3.6 β' . Note that both Figures 3.6 α' and 3.6 β' are in agreement with the results obtained in Section 3.10.1 where synthetic data with typical values for sensor's noise and measurement distortion were considered.



(α') Accuracy ($1/J_r$) of the presented algorithms using multiple datasets of real data from two different commercial magnetic field sensors.



(β') Robustness ($RB(\%)$) of the presented algorithms using multiple datasets of real data from two different commercial magnetic field sensors.

Figure 3.6: Performance characteristics of the presented algorithms using multiple datasets of real data from two different commercial magnetic field sensors.

3.11 Conclusion

To summarize, a complete and extensive study on calibration methods for low-cost magnetometers was carried out by the authors. Seven algorithms were selected for this purpose according to their popularity and their performance. A standard, unified and complete, linear measurement model was used here as the reference model for analyzing all calibration methods. After establishing the full calibration problem, these seven algorithms were discussed and were presented in an easy-to-implement way.

In order to evaluate fairly the presented algorithms' performance, we proposed a method for optimally generating artificial magnetometer data. This method was used for executing a plethora of Monte Carlo simulations. The evaluation metrics we focused on were the robustness, the accuracy and the efficiency of the algorithms. We designed several experiments to check the behavior of the algorithms under different values in bias, different values in non-orthogonality

errors, different number of measurements and finally under various orders of variance in noise. Finally, several datasets of real magnetometer's data, from two different, low-cost, commercial sensors were used to verify the results obtained using the artificial data.

The following summarizes our findings regarding the studied algorithms and their possible implementation. Except from the objective criteria that we established in Section 3.10 to evaluate and compare the presented algorithms (accuracy, robustness, computational efficiency), in Table 3.4 we also evaluate the algorithms in terms of simplicity. Simplicity is used as a (subjective) metric describing our personal experience developing and testing the algorithms. It is related both to the algorithmic complexity of the algorithms (which is not an inherent disadvantage) and the quality of their presentation in the original manuscripts. The algorithms are discussed in chronological order of publication.

- ✓ **TWOSTEP**: Extremely time efficient. Works effectively for small distortions. Has low accuracy in general. The method can be generalized to on-orbit calibration.
- ✓ **Crassidis et al.**: Easy to implement. Extremely time efficient. Works effectively for small to medium distortions. The method can be generalized to on-orbit calibration. It is the only algorithm that provides online update. It can be considered as a more accurate and effective version of TWOSTEP with similar time complexity.
- ✓ **Dorveaux et al.**: Easy to implement. Moderately time efficient. Robust and accurate, but vulnerable to large values of bias.
- ✓ **Vasconcelos et al.**: Difficult to implement. Characterized by high time-complexity. Exceptional accuracy and robustness for small distortions.
- ✓ **Ali et al.**: Robust and accurate. Very high computational cost. Some prior knowledge of the search space is beneficial. At the beginning of the algorithm, the unknown variables are randomized and thus it is not always ensured that the algorithm will reach an optimal point. Thus, a couple of repetitions might be needed. Using modern PSO algorithms which can constrain the search space and handle a few variable inequalities increases the algorithm's performance significantly.
- ✓ **Wu and Shi**: Difficult to implement. Characterized by high time-complexity. Exceptional accuracy even with larger distortion. We noticed a 1% failure of finding an initial estimate due to inadequacy of applying Cholesky decomposition.
- ✓ **MAG.I.C.AL**: Easy to implement. Moderately time efficient. Exceptional robustness and accuracy in both synthetic and real data experiments.

To conclude, in this work, we tried to cover a broad range of realistic cases and test the limits of the algorithms, noting that in real life the performance requirements differ from application

Algorithm	Simplicity	Robustness	Accuracy	Efficiency
TWOSTEP	✓✓	✓✓	✓	✓✓✓
Crassidis et al.	✓✓✓	✓✓	✓	✓✓✓
Dorveaux et al.	✓✓✓	✓✓✓	✓✓✓	✓✓
Vasconcelos et al.	✓	✓	✓✓	✓
Ali et al.	✓✓	✓✓✓	✓✓✓	✓
Wu and Shi	✓	✓	✓✓✓	✓
MAG.I.C.AL	✓✓✓	✓✓✓	✓✓✓	✓✓

Table 3.4: Algorithms' Comparison Summary

to another. In some applications computational efficiency may be of major importance while great accuracy may not be needed, while in others, a very accurate calibration is essential even if significantly more computation time is required for this. Thus there is no "perfect" algorithm appropriate for all applications; different algorithms may be more appropriate for different cases.

3.12 Appendix

3.12.1 Gradient Vector and Hessian Matrix for [30]

This section presents the algebraic expressions for the gradient and Hessian of the likelihood function (3.37), used in descent optimization methods. Let $u_k = y_k - h$, the gradient of the likelihood function J (3.35) is denoted by $\nabla J|_x = [\nabla J|_{\hat{T}} \nabla J|_h]$ and described by the submatrices

$$\nabla J|_{\hat{T}} = \sum_{k=0}^N \frac{2c_k}{\sigma^2} u_k \otimes \hat{T} u_k \quad (3.86\alpha')$$

$$\nabla J|_h = \sum_{k=0}^N \frac{-2c_k}{\sigma^2} \hat{T}^T \hat{T} u_k \quad (3.86\beta')$$

where $c_k = 1 - \|\hat{T} u_k\|^{-1}$. The Hessian

$$\nabla^2 J|_x = \begin{bmatrix} H_{\hat{T},\hat{T}} & H_{\hat{T},h} \\ H_{\hat{T},b}^T & H_{h,h} \end{bmatrix} \quad (3.87)$$

is given by the following submatrices

$$H_{\hat{T},\hat{T}} = \sum_{k=1}^N \frac{2}{\sigma^2} \left[\frac{(u_k u_k^T) \otimes (\hat{T} u_k u_k^T \hat{T}^T)}{\|\hat{T} u_k\|^3} + c_k [(u_k u_k^T) \otimes I_{3 \times 3}] \right] \quad (3.88\alpha')$$

$$H_{\hat{T},h} = \sum_{k=1}^N \frac{-2}{\sigma^2} \left[\frac{(u_k \otimes \hat{T} u_k) u_k^T \hat{T}^T \hat{T}}{\|\hat{T} u_k\|^3} + c_k [u_k \otimes \hat{T} + I_{3 \times 3} \otimes \hat{T} u_k] \right] \quad (3.88\beta')$$

$$H_{h,h} = \sum_{k=1}^N \frac{2}{\sigma^2} \left[\frac{\hat{T}^T \hat{T} u_k u_k^T \hat{T}^T \hat{T}}{\|\hat{T} u_k\|^3} + c_k \hat{T}^T \hat{T} \right] \quad (3.88\gamma')$$

These expressions can be found more analytically in the original paper [30], but are presented here as well since this paper provides a clear path to implementation.

3.12.2 Gradient Vector and Hessian Matrix for [31]

This section presents the algebraic expressions for the gradient and Hessian of the likelihood function (3.54), used in descent optimization methods. For notational simplicity \hat{T} and \hat{m} are replaced by T and m . Let $u_k = y_k - h$, the Jacobian vector and Hessian matrix can be respectively derived as

$$\nabla J|_x = \left[\begin{array}{cccc} \mathbf{J}_T^T & \mathbf{J}_h^T & \underbrace{\mathbf{J}_{m_k}^T}_{k=1:N} & \underbrace{\mathbf{J}_{\lambda_k}^T}_{k=1:N} \end{array} \right]^T \quad (3.89)$$

$$\nabla^2 J|_x = \left[\begin{array}{cccc} H_{TT} & H_{Th} & H_{Tm_k} \dots & 0_{9 \times 1} \dots \\ H_{Th}^T & H_{hh} & H_{hm_k} \dots & 0_{3 \times 1} \dots \\ H_{Tm_k}^T & H_{hm_k}^T & H_{m_k m_k} \dots & H_{m_k \lambda_k} \dots \\ \vdots & \vdots & \vdots & \vdots \\ 0_{9 \times 1}^T & 0_{3 \times 1}^T & H_{m_k \lambda_k}^T \dots & 0 \dots \\ \vdots & \vdots & \vdots & \vdots \end{array} \right] \quad (3.90)$$

where

$$\begin{aligned} \mathbf{J}_T &= -2 \sum_{k=1}^N m_k \otimes (u_k - T m_k) \\ \mathbf{J}_h &= -2 \sum_{k=1}^N (u_k - T m_k) \\ \mathbf{J}_{m_k} &= -2 T^T (u_k - T m_k) + 2 \lambda_k m_k \\ \mathbf{J}_{\lambda_k} &= \|m_k\|^2 - 1 \end{aligned} \quad (3.91)$$

and

$$\begin{aligned}
H_{TT} &= 2 \sum_{k=1}^N (m_k m_k^T) \otimes I \\
H_{Th} &= 2 \sum_{k=1}^N m_k \otimes I \\
H_{Tm_k} &= 2 ((m_k \otimes I) T - I \otimes (u_k - T m_k)) \\
H_{hh} &= 2NI \\
H_{hm_k} &= 2T \\
H_{m_k m_k} &= 2T^T T + 2\lambda_k I \\
H_{m_k \lambda_k} &= 2m_k
\end{aligned} \tag{3.92}$$

In [31], the calibration matrix, T , is considered to be an upper triangular matrix. Thus from both the gradient vector and the Hessian matrix, the rows and columns that correspond to the lower triangular elements of T must be removed.

These expressions are presented analytically in the original paper [31], but are presented here as well since this paper provides a clear path to implementation.

References

- [1] Abdelrahman Ali et al. "Swarm Optimization-Based Magnetometer Calibration for Personal Handheld Devices". In: *Sensors* 12.9 (2012), pp. 12455–12472. ISSN: 1424-8220. DOI: 10.3390/s120912455.
- [2] R Alonso and Shuster. "Complete Linear Attitude-Independent Magnetometer Calibration". In: *The Journal of the Astronautical Sciences*. 50 (2002), pp. 477–490.
- [3] R Alonso and Shuster. "TWOSTEP: A fast robust algorithm for attitude-independent magnetometer-bias determination." In: *The Journal of the Astronautical Sciences*. 50 (2002), pp. 433–451.
- [4] Guocan Cao, Xiang Xu, and Dacheng Xu. "Real-Time Calibration of Magnetometers Using the RLS/ML Algorithm". In: *Sensors* 20.2 (2020). ISSN: 1424-8220. DOI: 10.3390/s20020535. URL: <https://www.mdpi.com/1424-8220/20/2/535>.
- [5] John Crassidis, Kok-Lam Lai, and Richard R. Herman. "Real-Time Attitude-Independent Three-Axis Magnetometer Calibration". In: *JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS* 28 (Jan. 2005), pp. 115–120. DOI: 10.2514/1.6278.
- [6] John L. Crassidis. *Optimal Estimation of Dynamic Systems*. CRC Press, 2004.
- [7] E. Dorveaux et al. "Iterative calibration method for inertial and magnetic sensors". In: *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. Dec. 2009, pp. 8296–8303. DOI: 10.1109/CDC.2009.5399503.

-
- [8] Magnus Erik, Hvass Pedersen, and Hvass Laboratories. *Good Parameters for Particle Swarm Optimization By*. 2010.
- [9] C. C. Foster. “Elkaim, Extension of a two-step calibration methodology to include nonorthogonal sensor axes”. In: *IEEE Trans. on Aerospace and Electronic Systems* (2008), pp. 1070–1078.
- [10] B. Gambhir. *Determination of Magnetometer Biases Using Module RESIDG*. Tech. rep. Computer Sciences Corporation, 1975.
- [11] Demoz Gebre-Egziabher et al. “Calibration of Strapdown Magnetometers in Magnetic Field Domain”. In: *Journal of Aerospace Engineering - J AEROSP ENG* 19 (Apr. 2006). DOI: 10.1061/(ASCE)0893-1321(2006)19:2(87).
- [12] Alvaro Gonzalez. “Measurement of Areas on a Sphere Using Fibonacci and Latitude–Longitude Lattices”. In: *Mathematical geosciences* 42 (Jan. 2010), pp. 49–64. DOI: 10.1007/s11004-009-9257-x.
- [13] Neoclis Hadjigeorgiou et al. “Vector Magnetic Field Sensors: Operating Principles, Calibration and Applications”. In: *IEEE Sensors Journal* (2020), pp. 1–1. DOI: 10.1109/JSEN.2020.3045660.
- [14] Sabine Helwig, Juergen Branke, and Sanaz Mostaghim. “Experimental Analysis of Bound Handling Techniques in Particle Swarm Optimization”. In: *IEEE Transactions on Evolutionary Computation* 17.2 (2013), pp. 259–271. DOI: 10.1109/TEVC.2012.2189404.
- [15] Working Group 8 IAGA Division V. “Revision of International Geomagnetic Reference Field released”. In: *EOS Transactions* 77.16 (1996), pp. 153–153. DOI: 10.1029/96E000106.
- [16] J. Kennedy and R. Eberhart. “Particle swarm optimization”. In: *Proceedings of ICNN’95 - International Conference on Neural Networks*. Vol. 4. 1995, pp. 1942–1948. DOI: 10.1109/ICNN.1995.488968.
- [17] J. Kennedy, B.Gireesha Obaiahnahatti, and Yuhui Shi. “Swarm Intelligence”. In: *Morgan Kaufmann Academic Press, San Francisco* 1 (Jan. 2001), pp. 1931–1938.
- [18] M. Kok et al. “Calibration of a magnetometer in combination with inertial sensors”. In: *2012 15th International Conference on Information Fusion*. July 2012, pp. 787–793.
- [19] Manon Kok and Thomas B. Schön. “Maximum likelihood calibration of a magnetometer using inertial sensors”. In: *IFAC Proceedings Volumes* 47.3 (2014). 19th IFAC World Congress, pp. 92–97. ISSN: 1474-6670. DOI: <https://doi.org/10.3182/20140824-6-ZA-1003.02025>.
- [20] G.M LERNER. *Spacecraft Attitude Determination and Control*. Kluwer Academic Publishers, 1978.
- [21] Xiang Li and Zhi Li. “A new calibration method for tri-axial field sensors in strap-down navigation systems”. In: *Measurement Science and Technology* 23.10 (Sept. 2012), p. 105105. DOI: 10.1088/0957-0233/23/10/105105.

- [22] MATLAB. *Optimization Toolbox*. Natick, Massachusetts: The MathWorks Inc.
- [23] Efrén Mezura-Montes and Carlos Coello. “Constraint-handling in nature-inspired numerical optimization: Past, present and future”. In: *Swarm and Evolutionary Computation 1* (Dec. 2011), pp. 173–194. DOI: 10.1016/j.swevo.2011.10.001.
- [24] K. Papafotis and P. P. Sotiriadis. “Accelerometer and Magnetometer Joint Calibration and Axes Alignment”. In: *Technologies 8(1), 11* (Jan. 2020). DOI: 10.3390/technologies8010011.
- [25] K. Papafotis and P. P. Sotiriadis. “MAG.I.C.AL.—A Unified Methodology for Magnetic and Inertial Sensors Calibration and Alignment”. In: *IEEE Sensors Journal 19.18* (Sept. 2019), pp. 8241–8251. ISSN: 1530-437X. DOI: 10.1109/JSEN.2019.2919179.
- [26] Martin Roberts. *How to evenly distribute points on a sphere more effectively than the canonical Fibonacci Lattice*. <http://extremelearning.com.au/how-to-evenly-distribute-points-on-a-sphere-more-effectively-than-the-canonical-fibonacci-lattice/>.
- [27] P. Schonemann. “A generalized solution of the orthogonal procrustes problem”. In: *Psychometrika 31.1* (Mar. 1966), pp. 1–10.
- [28] John C. Springmann and James W. Cutler. “Attitude-Independent Magnetometer Calibration with Time-Varying Bias”. In: *Journal of Guidance, Control, and Dynamics 35.4* (2012), pp. 1080–1088. DOI: 10.2514/1.56726.
- [29] Gilbert Strang. *Linear Algebra and Its Applications*. Brooks Cole/Cengage Learning, 2007.
- [30] J. F. Vasconcelos et al. “Geometric Approach to Strapdown Magnetometer Calibration in Sensor Frame”. In: *IEEE Transactions on Aerospace and Electronic Systems 47.2* (Apr. 2011), pp. 1293–1306. ISSN: 0018-9251. DOI: 10.1109/TAES.2011.5751259.
- [31] Y. Wu and W. Shi. “On Calibration of Three-Axis Magnetometer”. In: *IEEE Sensors Journal 15.11* (Nov. 2015), pp. 6424–6431. ISSN: 1530-437X. DOI: 10.1109/JSEN.2015.2459767.
- [32] Y. Wu et al. “Dynamic Magnetometer Calibration and Alignment to Inertial Sensors by Kalman Filtering”. In: *IEEE Transactions on Control Systems Technology 26.2* (Mar. 2018), pp. 716–723. ISSN: 1063-6536. DOI: 10.1109/TCST.2017.2670527.

4

Multiple Accelerometers and Magnetometers Joint Calibration and Alignment

In this work we introduce a computationally efficient algorithm for simultaneous joint calibration and axes alignment of multiple 3-axis accelerometers or 3-axis magnetometers. The proposed algorithm applies to ensembles of sensors of the same type, fixed to the same rigid platform. It compensates for all linear time-invariant distortions such as scale-factor, cross-coupling, and offset, including the soft-iron and hard-iron distortions of the magnetometer. It can be implemented in a simple 12-step sequence and provides fast and accurate calibration without requiring any special piece of equipment.

4.1 Introduction

Inertial and magnetic sensors are widely used in a broad variety of applications, from low-cost commercial systems (smartphones, activity trackers, etc) to high-end marine, aerospace and military ones. Typical application fields include navigation [5], attitude estimation [14], image stabilization [4] and others. Expensive, calibrated, high accuracy sensors are used in many industrial and military applications. In commercial systems however, where cost is of ultimate importance, integrated, minimum-cost inertial and magnetic sensors are preferred.

Inertial sensors in micro-electro-mechanical (MEMS) form have become very popular over the past decades as they combine very small size with very low-cost. Their main drawback is their large error characteristics [8] that must be compensated if accuracy is needed. Integrated magnetic sensors also require calibration; even if their inherent accuracy is satisfactory, when embedded to a system, the so-called hard-iron and soft-iron distortions must be removed.

Hard-iron distortion is the additive magnetic field created by magnetic objects attached to the reference frame of the magnetic sensor. Soft-iron distortion is the alteration of the existing magnetic field caused by ferromagnetic materials attached to the sensor's frame. Both these distortions are usually caused by electronic components in the vicinity of the sensor or materials

used in the sensor's enclosure.

Note that post-production factory calibration requires using special calibration equipment and/or services, both of which dramatically raise the final cost of the sensors. Therefore factory calibration is not typically an option for low-cost systems. Thus when accuracy is important, a calibration procedure that can be applied without using any special piece of equipment is required.

As far as 3-axis accelerometers are concerned, most authors exploit the constant magnitude of the gravity for calibration purposes. More specifically, the calibration parameters are derived by measuring the sensor's output when it is placed in several different still positions. The calibration problem is either posed as a minimization problem [13, 3, 6, 17] or as an estimation one [7].

In a similar way, for 3-axis magnetometer calibration, the magnetic field of the earth is most commonly used as a reference. Using the fact that the magnitude of the measured magnetic field should be locally constant (away of magnetic disturbances), the calibration parameters are calculated by solving a minimization [2, 22, 12, 11, 21, 17] or an estimation [23] problem.

The similarity in the calibration approach of accelerometers and magnetometers, based on similar physical principles, allows developing and using a single algorithm to calibrate both of them.

While there are several efficient calibration algorithms for a single accelerometer or magnetometer, in many applications multiple sensors of the same kind are used to improve the measurement's accuracy [19]. In [18], the authors use a platform containing multiple magnetometers for orientation estimation. Six accelerometers arranged on a cube are used in [1] for a gyroscope-free inertial navigation application. In such cases, even if the sensors are individually calibrated, an extra step to align their sensitivity axes is required. This by itself is not a trivial step and requires multiple measurements and rotation matrices calculation.

In this work we expand the algorithm introduced for a single sensor's calibration in [17] to the case of multiple sensors. The proposed algorithm simultaneously addresses the calibration of multiple sensors and the alignment of their axes. It is based on a low computational cost iteration of a least squares problem providing fast convergence and accurate calibration. It is applied using a simple 12-step sequence of platform's orientations and requires no special piece of equipment. Simulation and experimental results prove the algorithm's performance and efficiency.

4.2 Sensors' Measurement Model and The Problem of Calibration

In this section we introduce the problem of simultaneous joint calibration and axes alignment of multiple 3-axis accelerometers or multiple 3-axis magnetometers. To do so, we first derive a general model relating the sensors' measurements with the true values of the specific force or the magnetic field respectively.

4.2.1 Accelerometer's Measurement Model

The measurement of an accelerometer is modeled as in [8] [16]

$$y_a = f + T_{sf}f + T_{cc}f + h_a + \varepsilon, \quad (4.1)$$

where y_a is the 3×1 measurement vector, f is the 3×1 true specific force vector, T_{sf} is the 3×3 diagonal matrix representing the scale-factor error, T_{cc} is the 3×3 matrix representing the cross-coupling error, h_a is the 3×1 accelerometer's offset vector and ε represents the 3×1 random noise vector.

Defining matrix $T_a \triangleq I_3 + T_{sf} + T_{cc}$, where I_3 is the 3×3 identity matrix, (4.1) can be written as

$$y_a = T_a f + h_a + \varepsilon \quad (4.2)$$

4.2.2 Magnetometer's Measurement Model

The measurement of a magnetometer is modeled following [22], [21], [10] and [15]

$$y_m = T_{sf}T_{cc}(T_{si}m + h_{hi}) + h_b + \varepsilon \quad (4.3)$$

where y_m is the 3×1 measurement vector, m is the 3×1 true magnetic field vector, T_{sf} denotes the 3×3 diagonal matrix representing the scale-factor error, T_{cc} is the 3×3 matrix representing the cross-coupling error, T_{si} is the 3×3 matrix modeling the soft-iron distortion, h_b is the 3×1 magnetometer's offset vector, h_{hi} is the 3×1 offset vector due to hard-iron distortion and ε denotes the 3×1 random noise vector.

Defining matrices $T_m \triangleq T_{sf}T_{cc}T_{si}$ and $h_m \triangleq T_{sf}T_{cc}h_{si} + h_b$, the magnetometer's measurement model becomes

$$y_m = T_m m + h_m + \varepsilon \quad (4.4)$$

4.2.3 Calibration as an Optimization Problem

The measurement models of the two sensor share the same form and thus we can use the general model of (4.5) to formulate both sensors calibration process as the same optimization problem.

$$y = Tn + h + \varepsilon \quad (4.5)$$

The purpose of the calibration is to estimate the 3×3 matrix T and the 3×1 vector h in order to minimize the measured noise ε strength.

The advantage of the proposed calibration method is that no special calibration equipment is needed, e.g., turn-table for the accelerometer or Gauss magnetic chamber and Maxwell coils setup for the magnetometer. This implies that the true specific force or the true magnetic field vectors are unknown.

Note however that the magnitude of the true specific force and that of the true magnetic field are fixed when the accelerometer is still and the magnetometer is in static and uniform magnetic field. Assuming that the calibration measurements were taken under these conditions, without loss of generality we assume that in (4.5) it is always $\|n\| = 1$. All norms in this paper are two-norms unless it is indicated otherwise.

Assuming K measurements y_1, y_2, \dots, y_K and writing (4.5) as $y_k = Tn_k + h + \varepsilon_k$, we define the cost function J_1 capturing the total error

$$J_1 = \sum_{k=1}^K \|\varepsilon_k\|^2 = \sum_{k=1}^K \|y_k - Tn_k - h\|^2 \quad (4.6)$$

and we form the optimization problem below whose solution is the calibration pair T, h and the field vectors $n_k, k = 1, 2, \dots, K$

$$\begin{aligned} & \underset{T, h, n_k}{\text{minimize}} && J_1 \\ & \text{subject to} && \|n_k\| = 1, \quad k = 1, 2, \dots, K \end{aligned} \quad (4.7)$$

Note that when (4.7) is used for a single sensor's calibration, the calibration matrix T is derived subject to orthogonal multiplication uncertainty. Specifically, if we replace T with TQ and n with $Q^T n$, where $Q \in O(3)$, the resulting measurement y is unaltered. Thus if multiple sensors are individually calibrated using (4.7), a misalignment between their sensitivity axes is expected.

To overcome this, we expand the single-sensor calibration algorithm (4.7) to (multiple) L sensors calibration algorithm incorporating the alignment of the sensors' axes as well. Let $n_k, k = 1, 2, \dots, K$ be the k^{th} unit field vector, simultaneously exciting all L sensors, and denote by $y_{\ell, k}$ and $\varepsilon_{\ell, k}$ the corresponding measurement and error of the ℓ^{th} sensor, $\ell = 1, 2, \dots, L$. Moreover, let T_{ℓ} and h_{ℓ} be the ℓ^{th} sensor's calibration matrix and offset vector respectively. We write (4.5) in the matrix form

$$Y = TN + H + E \quad (4.8)$$

where Y is the measurements matrix, E is the errors matrix

$$Y = \begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,K} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ y_{L,1} & y_{L,2} & \cdots & y_{L,K} \end{bmatrix}, \quad E = \begin{bmatrix} \varepsilon_{1,1} & \varepsilon_{1,2} & \cdots & \varepsilon_{1,K} \\ \varepsilon_{2,1} & \varepsilon_{2,2} & \cdots & \varepsilon_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon_{L,1} & \varepsilon_{L,2} & \cdots & \varepsilon_{L,K} \end{bmatrix}$$

T and H are the calibration matrices in block form

$$\begin{aligned} T &= \begin{bmatrix} T_1^T & T_2^T & \cdots & T_L^T \end{bmatrix}^T \\ H &= \begin{bmatrix} h_1^T & h_2^T & \cdots & h_L^T \end{bmatrix}^T \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \end{aligned}$$

and N is the set of true specific force or magnetic field vectors, common to all sensors

$$N = \begin{bmatrix} n_1 & n_2 & \dots & n_K \end{bmatrix}$$

As in the case of a single sensor, the cost function capturing the total error is expressed as

$$J_L = \sum_{\ell=1}^L \sum_{k=1}^K \|\varepsilon_{\ell,k}\|^2 = \|E\|_F^2 = \text{tr}(E^T E) \quad (4.9)$$

where the subscript F denotes the Frobenius norm [9]. The corresponding optimization problem becomes

$$\begin{aligned} & \underset{T, H, N}{\text{minimize}} && J_L \\ & \text{subject to} && \|n_k\| = 1, \quad k = 1, 2, \dots, K \end{aligned} \quad (4.10)$$

Note that in (4.10), the set of unit-magnitude vectors n_k is common for all sensors, resulting in the alignment of their axes.

4.3 The Proposed Algorithm

A typical way to solve optimization problems like (4.10) is by using the gradient descent or the Newton-Raphson methods. However, both of them methods depend strongly on a good initial estimate of the unknowns in order to converge. Especially in the magnetometer's case, finding an initial estimate is not trivial due to the uncertainty of soft-iron and hard-iron distortions reflecting on more challenging structures of T and H . In this work we propose a computationally efficient, least-squares based, iterative algorithm for solving (4.10) and achieving joint calibration and axes alignment of multiple sensors.

4.3.1 Algorithm Description

In order to solve (4.10), we form a two-step algorithm. In the first step, we use an estimation of the calibration matrices T and H in order to solve for matrix N . Then, using the new matrix N , we improve our estimation of T and H while forcing the magnitude of the true specific force or magnetic field to be equal to one.

Given a set of K measurements, every iteration begins by calculating N using the estimate of T and H from the previous one. Assuming T if of full rank, we solve, (4.8) using the least-squares method [20]

$$\tilde{N} = (T^T T)^{-1} T^T (Y - H). \quad (4.11)$$

To ensure that the true specific force or the true magnetic field vectors n_k , $k = 1, 2, \dots, K$ are of unit norm, we form the new estimate of N using the column vectors \tilde{n}_k , $k = 1, 2, \dots, K$ of \tilde{N} as follows

$$N = \begin{bmatrix} \frac{\tilde{n}_1}{\|\tilde{n}_1\|} & \frac{\tilde{n}_2}{\|\tilde{n}_2\|} & \dots & \frac{\tilde{n}_K}{\|\tilde{n}_K\|} \end{bmatrix} \quad (4.12)$$

Using this estimate of N , we rewrite (4.8) as

$$Y = XG + E \quad (4.13)$$

where

$$X = \begin{bmatrix} T_1^T & T_2^T & \dots & T_L^T \\ h_1^T & h_2^T & \dots & h_L^T \end{bmatrix}^T, \quad G = \begin{bmatrix} N \\ 1_{1 \times K} \end{bmatrix}$$

The updated T and H are derived by solving (4.13) for X in the least-squares sense. Assuming G is of full rank, we get

$$X = YG^T(GG^T)^{-1} \quad (4.14)$$

The proposed algorithm is summarized in Algorithm 13. It is initiated using $T_1 = T_2 = \dots = T_L = I_3$ and $h_1 = h_2 = \dots = h_L = 0_{3 \times 1}$.

Algorithm 13: Proposed Algorithm

- Step 1: Initialize $T_l = I_3$ and $h_l = 0_{3 \times 1}$, $l = 1, 2, \dots, L$
 - Step 2: Calculate \tilde{N} using (4.11) and form N as in (4.12)
 - Step 3: Formulate matrices X and G according to (4.13)
 - Step 4: Calculate X using (4.14)
 - Step 5: Extract T and H from X
 - Step 6: Calculate J_1 , for every sensor according to (4.6)
 - Step 7: Repeat steps 2-6 until J_1 is sufficiently small for every sensor
-

The convergence of the algorithm could be evaluated in every iteration by using the cost function J_L . However, J_L does not provide any information about the calibration of each individual sensor. Thus we prefer to use the cost function J_1 , evaluated for every sensor, as a metric of convergence.

4.3.2 Measurements Acquisition Procedure

The proposed algorithm jointly calibrates and aligns the axes of multiple accelerometers or magnetometers when they are all mounted on the same rigid platform. To that purpose we recommend the 12-step measurement acquisition procedure introduced in [17].

4.4 Experimental and Simulation Results

The proposed algorithm's performance is evaluated using both sensors' measurements, from two 3-axis magnetometers mounted on the same rigid platform, and simulated data of a large number of sensors ($L = 12$). Both measured and simulated data follow the measurement acquisition procedure introduced in [17].

In Figures 4.1 and 4.2 the algorithm's convergence using measured and simulated data respectively is presented. In both cases, the algorithm converges to a small value of the cost function in a way that appears to be monotonic.

In Figure 4.1 the fractional mean magnitude error as a function of the iteration is also shown as a metric of the algorithm's accuracy. Note that it becomes significantly small after a few iterations, even before the cost function converges. This fact, in combination with the small computational burden of each iteration demonstrate the computational efficiency of the proposed algorithm.

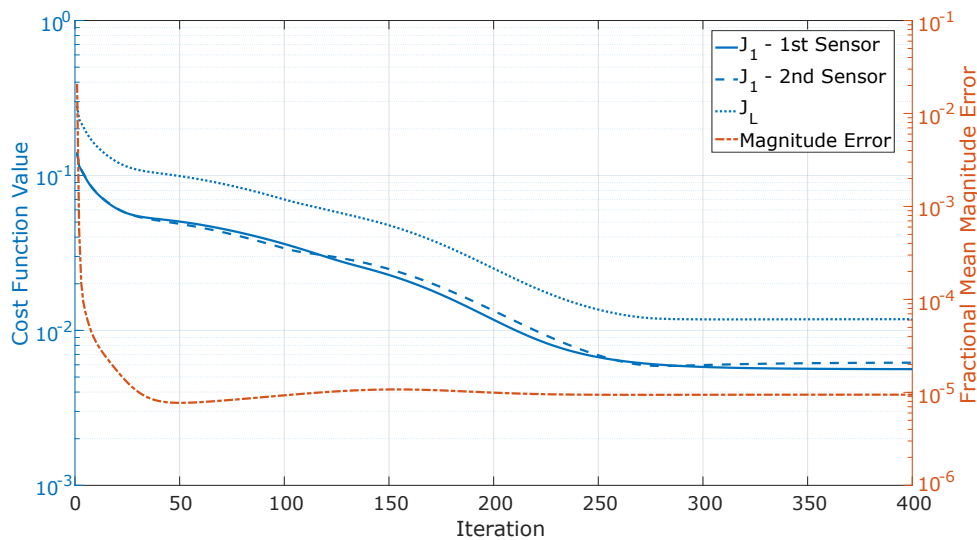


Figure 4.1: Algorithm convergence using measured data - Fractional mean magnitude error.

In Figure 4.3 the raw and calibrated magnetometers' measurements magnitude while the measurement platform is rotated in five different orientations are presented. As the measurement took place away from any magnetic disturbance, the measured magnetic field should be of constant magnitude. As seen in Figure 4.3, the magnitude of the raw measurements changes when the sensor is rotated while the magnitude of the calibrated data is constant and independent of the platform's orientation.

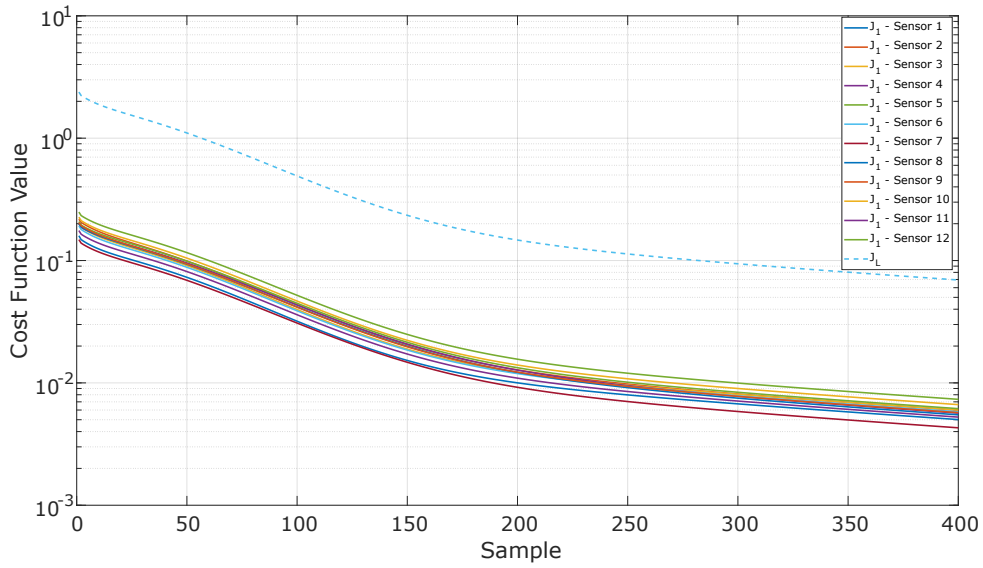


Figure 4.2: Algorithm convergence using simulated data.

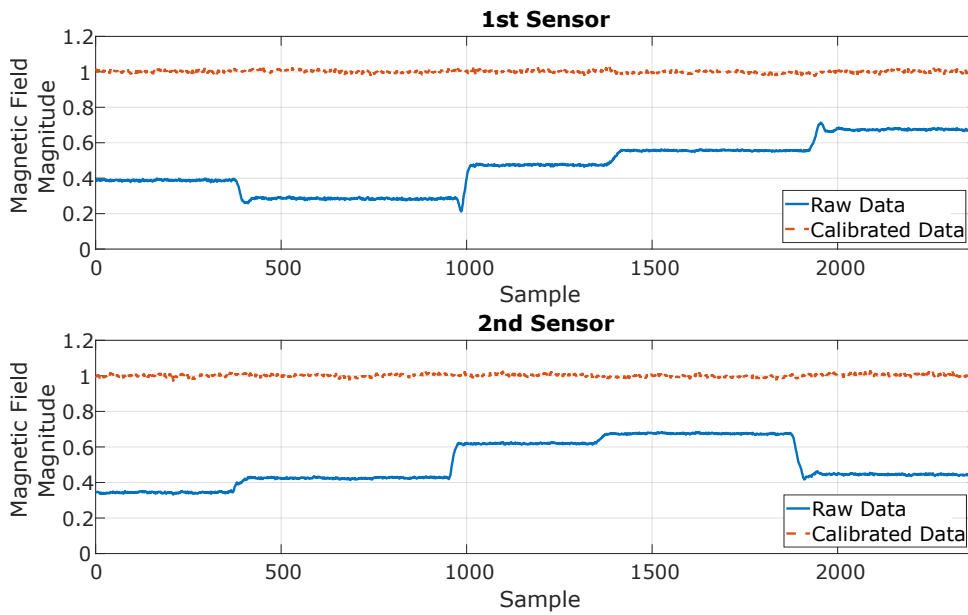


Figure 4.3: Raw and calibrated magnetometer's measurements magnitude in different orientations.

References

- [1] E. Akeila, Z. Salcic, and A. Swain. "Implementation, calibration and testing of GFINS models based on six-accelerometer cube". In: *TENCON 2008 - 2008 IEEE Region 10 Conference*. Nov. 2008, pp. 1–6. DOI: 10.1109/TENCON.2008.4766567.

-
- [2] R. Alonso and Shuster. “TWOSTEP: A fast robust algorithm for attitude-independent magnetometer-bias determination.” In: *The Journal of the Astronautical Sciences*. 50 (2002), pp. 433–451.
- [3] N. Ammann, A. Derksen, and C. Heck. “A novel magnetometer-accelerometer calibration based on a least squares approach”. In: *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. June 2015, pp. 577–585. DOI: 10.1109/ICUAS.2015.7152338.
- [4] C. Chu. “Video stabilization for stereoscopic 3D on 3D mobile devices”. In: *2014 IEEE International Conference on Multimedia and Expo (ICME)*. July 2014, pp. 1–6. DOI: 10.1109/ICME.2014.6890150.
- [5] C. Fischer, P. Talkad Sukumar, and M. Hazas. “Tutorial: Implementing a Pedestrian Tracker Using Inertial Sensors”. In: *IEEE Pervasive Computing* 12.2 (Apr. 2013), pp. 17–27. ISSN: 1536-1268. DOI: 10.1109/MPRV.2012.16.
- [6] I. Frosio, F. Pedersini, and N. A. Borghese. “Autocalibration of MEMS Accelerometers”. In: *IEEE Transactions on Instrumentation and Measurement* 58.6 (June 2009), pp. 2034–2041. ISSN: 0018-9456. DOI: 10.1109/TIM.2008.2006137.
- [7] M. Glueck, A. Buhmann, and Y. Manoli. “Autocalibration of MEMS accelerometers”. In: *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*. May 2012, pp. 1788–1793. DOI: 10.1109/I2MTC.2012.6229157.
- [8] Paul D. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2013.
- [9] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2013.
- [10] M. Kok et al. “Calibration of a magnetometer in combination with inertial sensors”. In: *2012 15th International Conference on Information Fusion*. July 2012, pp. 787–793.
- [11] Manon Kok and Thomas B. Schon. “Magnetometer Calibration Using Inertial Sensors”. In: *IEEE Sensors Journal* 16.14 (July 2016), pp. 5679–5689. ISSN: 1530-437X. DOI: 10.1109/JSEN.2016.2569160.
- [12] Manon Kok and Thomas B. Schön. “Maximum likelihood calibration of a magnetometer using inertial sensors”. In: *IFAC Proceedings Volumes* 47.3 (2014). 19th IFAC World Congress, pp. 92–97. ISSN: 1474-6670. DOI: <https://doi.org/10.3182/20140824-6-ZA-1003.02025>.
- [13] X. Lu, Z. Liu, and J. He. “Maximum Likelihood Approach for Low-Cost MEMS Triaxial Accelerometer Calibration”. In: *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. Vol. 01. Aug. 2016, pp. 179–182. DOI: 10.1109/IHMSC.2016.184.

- [14] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. “Estimation of IMU and MARG orientation using a gradient descent algorithm”. In: *2011 IEEE International Conference on Rehabilitation Robotics*. June 2011, pp. 1–7. DOI: 10.1109/ICORR.2011.5975346.
- [15] J Metge et al. “Calibration of an inertial-magnetic measurement unit without external equipment, in the presence of dynamic magnetic disturbances”. In: *Measurement Science and Technology* 25.12 (2014), p. 125106. URL: <http://stacks.iop.org/0957-0233/25/i=12/a=125106>.
- [16] Aboelmagd Noureldin, Tashfeen B. Karamat, and Jacques Georgy. *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*. Springer-Verlag Berlin Heidelberg, 2013.
- [17] K. Papafotis and P. P. Sotiriadis. “MAG.I.C.AL.—A Unified Methodology for Magnetic and Inertial Sensors Calibration and Alignment”. In: *IEEE Sensors Journal* 19.18 (Sept. 2019), pp. 8241–8251. ISSN: 1530-437X. DOI: 10.1109/JSEN.2019.2919179.
- [18] V. Renaudin, M. H. Afzal, and G. Lachapelle. “New method for magnetometers based orientation estimation”. In: *IEEE/ION Position, Location and Navigation Symposium*. May 2010, pp. 348–356. DOI: 10.1109/PLANS.2010.5507301.
- [19] I. Skog, J. Nilsson, and P. Händel. “An open-source multi inertial measurement unit (MIMU) platform”. In: *2014 International Symposium on Inertial Sensors and Systems (ISISS)*. Feb. 2014, pp. 1–4. DOI: 10.1109/ISISS.2014.6782523.
- [20] Gilbert Strang. *Linear Algebra and Its Applications*. Brooks Cole/Cengage Learning, 2007.
- [21] J. F. Vasconcelos et al. “Geometric Approach to Strapdown Magnetometer Calibration in Sensor Frame”. In: *IEEE Transactions on Aerospace and Electronic Systems* 47.2 (Apr. 2011), pp. 1293–1306. ISSN: 0018-9251. DOI: 10.1109/TAES.2011.5751259.
- [22] Y. Wu and W. Shi. “On Calibration of Three-Axis Magnetometer”. In: *IEEE Sensors Journal* 15.11 (Nov. 2015), pp. 6424–6431. ISSN: 1530-437X. DOI: 10.1109/JSEN.2015.2459767.
- [23] Y. Wu et al. “Dynamic Magnetometer Calibration and Alignment to Inertial Sensors by Kalman Filtering”. In: *IEEE Transactions on Control Systems Technology* 26.2 (Mar. 2018), pp. 716–723. ISSN: 1063-6536. DOI: 10.1109/TCST.2017.2670527.

Part 2

Applications of Inertial and Magnetic Field Sensors

5

Hardware Design of a Low-Cost IMU

An inertial measurement unit (IMU) based on low-cost inertial sensors was designed for experimental purposes. The designed system embeds the STMicroelectronics LSM9DS1 system-in-package (SiP) which contains a 3-axis accelerometer, a 3-axis gyroscope and a 3-axis magnetometer. The data are handled by a 32-bit microcontroller (STMicroelectronics STM32F746) and stored in a flash memory (temporarily) and a SD card (permanently). The system is powered by a standard 3.7V Li-ion battery. A system-level diagram is shown in Figure 5.1.

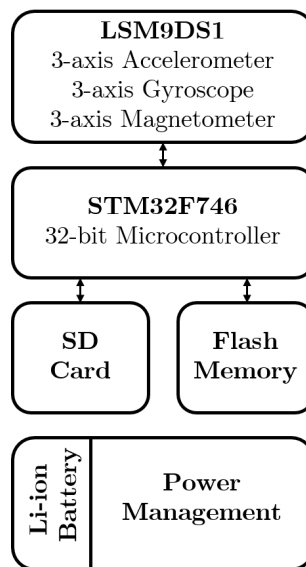


Figure 5.1: System-level diagram of the designed IMU.

The basic performance characteristics of the LSM9DS1 are presented in Tables 5.1, 5.2 and 5.3. The operating parameters of the three sensors (output data rate, measurement range etc.) can be configured via the embedded USB serial port.

	Accelerometer
Measurement Range	$\pm 2g - \pm 16g$
Output Resolution	16bits
Sensitivity	0.732mg/LSB ($\pm 16g$)
Output Data Rate (max)	952Hz

Table 5.1: Basic performance characteristics of the LSM9DS1 SiP (Accelerometer).

	Gyroscope
Measurement Range	$\pm 245dps - \pm 2000dps$
Output Resolution	16bits
Sensitivity	70mdps/LSB ($\pm 2000dps$)
Output Data Rate (max)	952Hz

Table 5.2: Basic performance characteristics of the LSM9DS1 SiP (Gyroscope).

	Magnetometer
Measurement Range	$\pm 4Gauss - \pm 16Gauss$
Output Resolution	16bits
Sensitivity	0.58mGauss/LSB ($\pm 16Gauss$)
Output Data Rate (max)	80Hz

Table 5.3: Basic performance characteristics of the LSM9DS1 SiP (Magnetometer).

The designed IMU is shown in Figure 5.2. The complete schematics of the designed IMU are presented in Appendix 10.

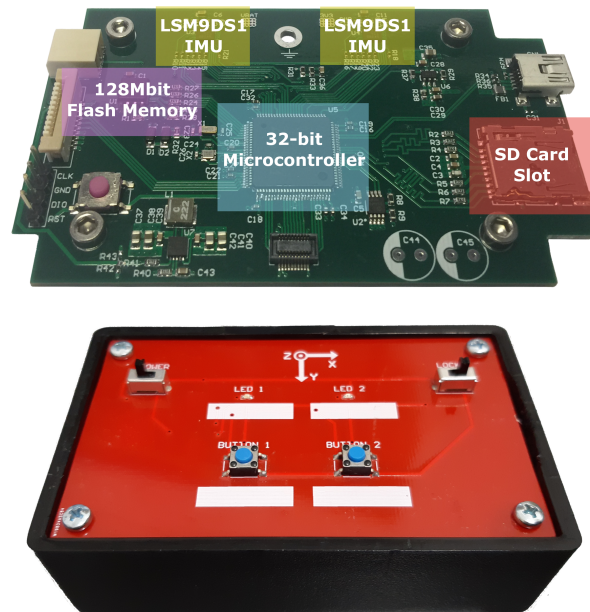


Figure 5.2: The designed IMU printed circuit board (up) and enclosure (down).

6

The Importance of Sensors' Calibration in Inertial Navigation Systems

In this work, we explore the importance of sensors' calibration in inertial navigation applications. We focus on the case of low-cost systems, typically using MEMS inertial sensors, where the extra calibration cost is a critical parameter. We highlight the importance of calibration by deriving a bound of the evolution of the attitude and velocity error as a function of the calibration parameters' error. Then, we use low-cost 3-axis accelerometer and 3-axis gyroscope along with a popular pedestrian inertial navigation algorithm to experimentally confirm that raw sensor's data can be highly inappropriate for navigation purposes. Finally, we use the MAG.I.C.AL methodology for joint calibration and axes alignment of inertial and magnetic sensors to achieve high accuracy measurements resulting in a reliable inertial navigation system.

6.1 Introduction

Satellite-based systems (GPS, Galileo, GLONASS etc.) are the dominant navigation technology. Even though they provide sufficiently accurate navigation for most applications, they all come with the same drawbacks: they have limited refresh rate, they don't work in indoor environments and they are susceptible to jamming. To overcome these limitations, several alternative navigation technologies have been developed during the past decades. The concurrent development of the micro-electro-mechanical systems (MEMS) led to a significant growth of inertial navigation systems.

Inertial navigation systems (INS) are based on inertial sensors (accelerometers and gyroscopes) to calculate the velocity, orientation and position of a moving object. They are commonly used in a wide range of applications, from low-cost commercial systems, to high-end military, marine and aerospace applications. Although INS yield accurate short-term navigation, their long-term performance is degraded, mainly due to the heading error caused by gyroscope's noise and offset

drift [6]. To improve the long-term performance, some authors combine INS with other navigation technologies (mostly satellite or RF based [12, 10]) while others use additional sensors (usually a magnetometer [6], [14]) to correct the estimated heading.

In the case of low-cost systems, MEMS inertial sensors are usually preferred due to their significantly lower cost and small size. However, a major disadvantage of MEMS inertial sensors is their large error characteristics [6]. Thus, in order to use them in an INS, a calibration procedure that compensates for the deterministic part of their error is required. In addition, the combined use of the accelerometer's, gyroscope's and maybe magnetometer's data gives rise to the need of alignment between the axes of the three sensors.

Although sensors' calibration and alignment are of major importance for an accurate INS, existing works take them for granted and only deal with the development of the navigation algorithms. Specifically, in [7, 8, 9, 5, 2, 3] expensive, already calibrated, commercially available sensor modules are used to evaluate the proposed algorithms. The authors in [12, 14, 15, 11] use custom sensor modules to evaluate the proposed INS but don't provide any details about the sensors' calibration and axes alignment.

Especially when low-cost systems are concerned, sensors' calibration and alignment could determine the overall system's cost. In the case of MEMS sensors, factory calibration is not an option as it would raise the sensors' cost significantly. In addition, standard after-production calibration and alignment techniques require expensive equipment (like a turn-table) that would also raise the overall system's cost.

In this work we derive the attitude and velocity error propagation equations as a function of accelerometer's and gyroscope's calibration parameters. Then, we design a low-cost inertial measurement unit (IMU) consisting of a MEMS 3-axis accelerometer and a MEMS 3-axis gyroscope. We calibrate the inertial sensors using the recently introduced MAG.I.C.AL. methodology [13] for joint calibration and axes alignment of inertial and magnetic sensors, and show how the raw sensors' data result in large attitude and velocity error. Finally, we use both the raw and calibrated sensors' data along with a popular pedestrian navigation algorithm, to experimentally demonstrate how the large error characteristics of the MEMS sensors affect the navigation accuracy.

6.2 Inertial Sensors' Error Propagation In The Inertial Frame

In this section the propagation of inertial sensors' error when they are used in navigation applications is described. In particular, we derive the error propagation equations in the inertial frame for attitude and velocity as a function of calibration parameters (T_a, h_a) and (T_g, h_g) of the accelerometer and gyroscope respectively.

6.2.1 Notation

Several different notations are used in the literature to describe kinematic quantities. In our analysis we will use the notation of [6]. More specifically, any kinematic quantity x , such as acceleration, velocity, position or angular velocity, is denoted as follows.

$$x_{\beta\alpha}^\gamma$$

where α is the body frame, β is the reference frame and γ is the resolving frame. In addition, the frame transformation matrix which transforms the resolving frame from α to β is denoted as C_α^β .

Also note that in the rest of this work, the true value of any quantity q is denoted as \tilde{q} while the measured one is denoted as q .

6.2.2 Attitude Error Propagation

The attitude error in the inertial frame is defined as:

$$\delta C_b^i = \tilde{C}_b^i C_i^b \quad (6.1)$$

where \tilde{C}_b^i is the true attitude while C_i^b is the attitude measured by the gyroscope. The differentiation of (6.1) yields

$$\delta \dot{C}_b^i = \dot{\tilde{C}}_b^i C_i^b + \tilde{C}_b^i \dot{C}_i^b \quad (6.2)$$

The time derivative of a coordinate transformation matrix is

$$\dot{C}_\alpha^\beta = C_\alpha^\beta \Omega_{\beta\alpha}^\alpha \quad (6.3)$$

where $\Omega_{\beta\alpha}^\alpha$ is the cross-product matrix of the angular velocity vector $\omega_{\beta\alpha}^\alpha$. Substituting the derivatives in (6.2), we get

$$\delta \dot{C}_b^i = \tilde{C}_b^i \delta \Omega \tilde{C}_i^b \delta C_b^i \quad (6.4)$$

where $\delta \Omega$ is the difference between the cross-product matrix of the true value of angular velocity vector $\tilde{\Omega}_{ib}^b$ and the one of the measured by the gyroscope angular velocity vector Ω_{ib}^b .

In our analysis we want to express $\delta \dot{C}_b^i$ as a function of the calibration parameters T_g and

h_g . To that purpose, we express $\delta\Omega$ as a function of the gyroscope's measurement vector ω_{ib}^b :

$$\delta\Omega = \begin{bmatrix} P_1\delta\omega & P_2\delta\omega & P_3\delta\omega \end{bmatrix} - \text{diag}(\delta\omega) \quad (6.5)$$

where $\delta\omega = (I_3 - T_g)\tilde{\omega} - h_g$, $P_1 = \begin{bmatrix} e_1 & e_3 & -e_2 \end{bmatrix}^T$, $P_2 = \begin{bmatrix} -e_3 & e_2 & e_1 \end{bmatrix}^T$ and $P_3 = \begin{bmatrix} e_2 & -e_1 & e_3 \end{bmatrix}^T$. Note that e_k is the k^{th} normal vector in \mathbb{R}^K .

The evolution of the attitude error in time is

$$\delta C_b^i(t) = \int_0^t \delta C_b^i(\tau) d\tau + \delta C_b^i(0) \quad (6.6)$$

Given that $\delta C_b^i(0) = I_3$ we write

$$\|\delta C_b^i(t) - I_3\| \leq \int_0^t \|\delta C_b^i(\tau)\| d\tau \quad (6.7)$$

Taking the Frobenius norm of (6.4) we get

$$\|\dot{\delta C}_b^i\|_F \leq 15\|\delta\omega\|_2 \quad (6.8)$$

Using (6.7) and (6.8), we write

$$\|\delta C_b^i(t) - I_3\| \leq 15t (\|I_3 - T_g\| w_B + \|h_g\|) \quad (6.9)$$

where w_B is a bound for the angular velocity magnitude and depends on the application.

6.2.3 Velocity Error Propagation

The velocity error is defined as

$$\delta V = \tilde{V} - V \quad (6.10)$$

The derivative of (6.10) is

$$\delta \dot{V} = \tilde{C}_b^i \tilde{f} - \delta C_b^i \tilde{C}_b^i (T_a \tilde{f} + h_a) \quad (6.11)$$

The evolution of the velocity error in time is derived by a similar analysis to that of the attitude error.

$$\|\delta V(t)\| \leq t [f_b (\|I_3 - T_a\| + \|T_a\| \|\delta C_b^i(t) - I_3\|) + \|h_a\|] \quad (6.12)$$

As seen in (6.12), the velocity error depends on both the accelerometer's and gyroscope's errors.

6.3 Experimental Results

6.3.1 Pedestrian Inertial Navigation Using Shoe-Mounted Inertial Sensors

A common inertial navigation application is the pedestrian navigation with shoe-mounted inertial sensors. In this case, a 3-axis accelerometer and a 3-axis gyroscope are mounted on the shoe of a walking human. Using their data, the velocity, orientation and position are calculated.

In such applications, the zero velocity update (ZUPT) method is typically used [4] [5]. ZUPT method is based on the fact that during the stance phase of the human walking [2], the velocity of the shoe is zero. This information is usually used as input to a Kalman filter [4] [5] in order to correct the error of the velocity, orientation and position estimations.

In this work, we use the pedestrian navigation algorithm proposed in [4] to explore the effect of sensors' calibration on the navigation accuracy. The proposed algorithm in [4] provides accurate navigation using shoe-mounted inertial sensors and zero velocity updates in a Kalman filter architecture.

6.3.2 Experiment Procedure

Using a low-cost IMU, we recorded accelerometer's and gyroscope's measurements of a walking human. More specifically a 290m walk around the campus football court was recorded while the IMU was mounted on the shoe. In order to explore the importance of sensors' calibration, we reconstructed the walked trajectory using a) uncalibrated inertial sensors, b) calibrated accelerometer and offset compensated gyroscope and c) fully calibrated accelerometer and gyroscope.

For the inertial sensors' calibration, we exploited the recently introduced MAG.I.C.AL. methodology. MAG.I.C.AL. provides unified calibration and joint axes alignment of 3-axis magnetometer, 3-axis accelerometer and 3-axis gyroscope. MAG.I.C.AL. compensates for all linear time-invariant distortions such as scale-factor, cross-coupling and offset, including the soft-iron and hard-iron distortion of the magnetometer. It is applied in a simple 15-step sequence of approximate placements and rotations of the sensors, made by hand, without requiring any special piece of equipment.

6.3.3 Error Propagation

In Section 6.2 a bound for the evolution of attitude error as a function of accelerometer's and gyroscope's calibration parameters is derived. Using the calibration parameters calculated using MAG.I.C.AL. methodology and the analysis of Section 6.2 we can explore the effect of calibration on the attitude and velocity error propagation.

In Figure 6.1 the evolution of attitude error in time is presented. As seen in Figure 6.1 the attitude error rises significantly after a few seconds when using uncalibrated sensors. Accelerometer's calibration and gyroscopes offset compensation improves the error evolution significantly but eventually large attitude errors are accumulated.

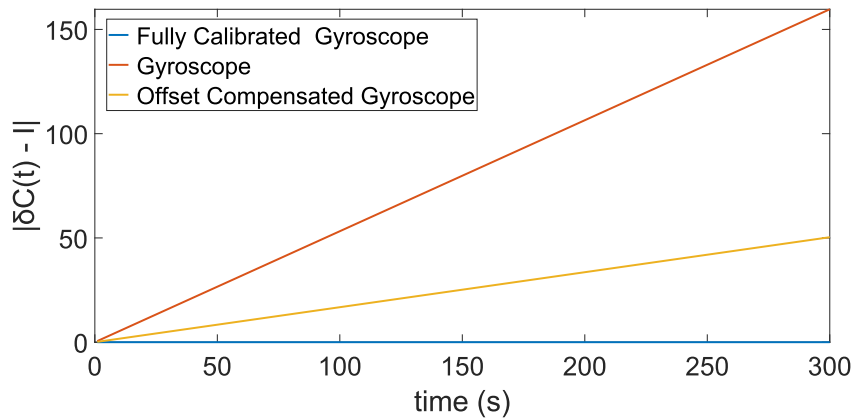


Figure 6.1: Evolution of attitude error in time.

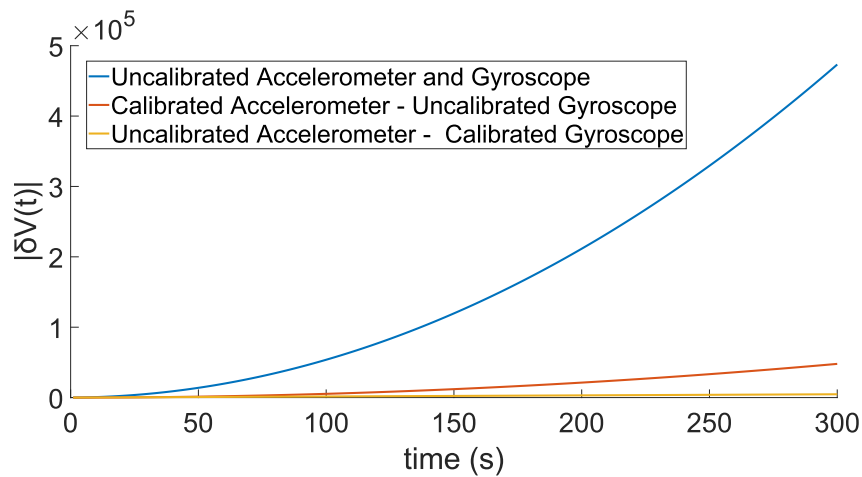


Figure 6.2: Evolution of velocity error in time.

In Figure 6.2 the evolution of velocity error in time is shown. As in the case of attitude, the velocity error also rises significantly after a few seconds when using uncalibrated sensors. Also note that, according to Figure 6.2, the gyroscope's error is the dominant error factor in the velocity error.

6.3.4 Trajectory reconstruction

The reconstructed trajectory using raw sensor's data is depicted in Figure 6.3 α' . As seen in Figure 6.3 α' , raw sensor's data are highly inappropriate for navigation purposes.

For accelerometer's calibration, there are several easy-to-apply methods without requiring any special piece of equipment [1, 13, 16]. In addition, although gyroscope's calibration is not trivial without using appropriate equipment, it's offset is easy to remove as it is just the sensor's output while it is still. By doing so, the navigation results are significantly improved as shown in Figure 6.3 β' .

In Figure 6.3 γ' the reconstructed trajectory using calibrated inertial sensors is presented.

The trajectory shown in Figure 6.3 γ' , exhibit a position error of about 5m in a 290m walk. The resulted navigation performance may not be state-of-the-art but it is actually impressive considering that we used very low-cost sensors and no special calibration equipment. In addition, the navigation algorithm used in this work is a basic algorithm using only inertial sensors. More complicated algorithms as well as the use of extra sensors (such as magnetometer) would provide even smaller position error.

Note that as seen in both Figures 6.3 β' and 6.3 γ' the GPS sampling rate is quite low causing sharp corners in the reconstructed trajectory.

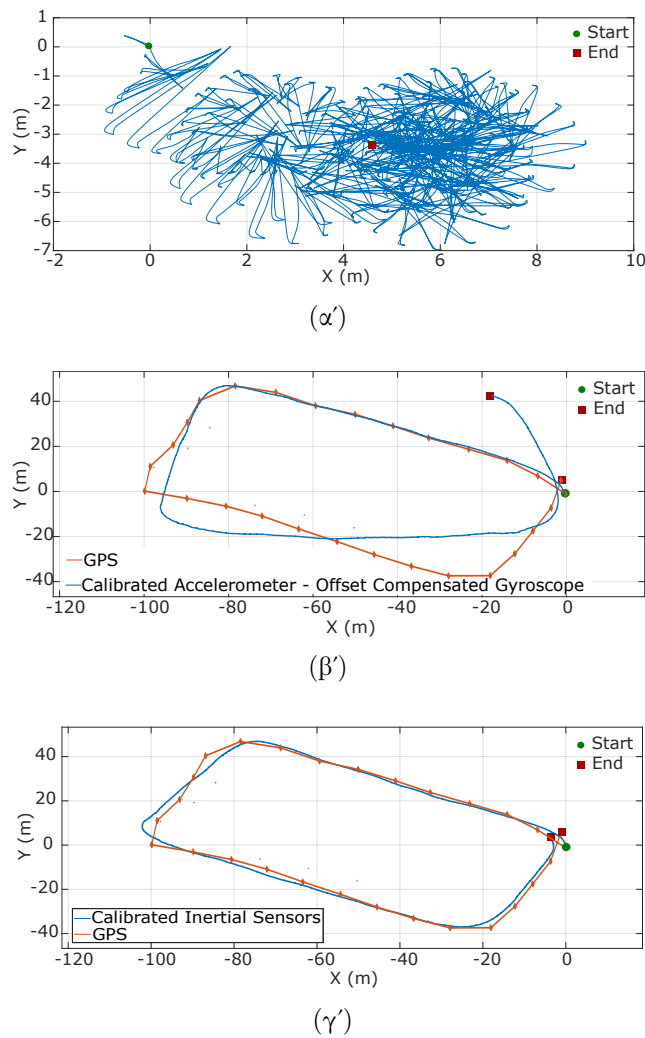


Figure 6.3: Reconstructed trajectory using a) uncalibrated inertial sensors, b) calibrated accelerometer and offset compensated gyroscope and c) fully calibrated inertial sensors

References

- [1] N. Ammann, A. Derksen, and C. Heck. "A novel magnetometer-accelerometer calibration based on a least squares approach". In: *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. June 2015, pp. 577–585. DOI: 10.1109/ICUAS.2015.7152338.
- [2] O. Bebek et al. "Personal Navigation via High-Resolution Gait-Corrected Inertial Measurement Units". In: *IEEE Transactions on Instrumentation and Measurement* 59.11 (Nov. 2010), pp. 3018–3027. ISSN: 0018-9456. DOI: 10.1109/TIM.2010.2046595.
- [3] Raul Feliz, Eduardo Zalama, and Jaime Gómez-García-Bermejo. "Pedestrian Tracking Using Inertial Sensors". In: *Journal of Physical Agents* 3 (Jan. 2009). DOI: 10.14198/JoPha.2009.3.1.05.
- [4] C. Fischer, P. Talkad Sukumar, and M. Hazas. "Tutorial: Implementing a Pedestrian Tracker Using Inertial Sensors". In: *IEEE Pervasive Computing* 12.2 (Apr. 2013), pp. 17–27. ISSN: 1536-1268. DOI: 10.1109/MPRV.2012.16.
- [5] E. Foxlin. "Pedestrian tracking with shoe-mounted inertial sensors". In: *IEEE Computer Graphics and Applications* 25.6 (Nov. 2005), pp. 38–46. ISSN: 0272-1716. DOI: 10.1109/MCG.2005.140.
- [6] Paul D. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2013.
- [7] Hang Guo et al. "Indoor positioning based on foot-mounted IMU". In: *Bulletin of the Polish Academy of Sciences Technical Sciences* 63 (Sept. 2015). DOI: 10.1515/bpasts-2015-0074.
- [8] A. R. Jimenez et al. "A comparison of Pedestrian Dead-Reckoning algorithms using a low-cost MEMS IMU". In: *2009 IEEE International Symposium on Intelligent Signal Processing*. Aug. 2009, pp. 37–42. DOI: 10.1109/WISP.2009.5286542.
- [9] A. R. Jiménez et al. "Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU". In: *2010 7th Workshop on Positioning, Navigation and Communication*. Mar. 2010, pp. 135–143. DOI: 10.1109/WPNC.2010.5649300.
- [10] R. Jirawimut et al. "A method for dead reckoning parameter correction in pedestrian navigation system". In: *IMTC 2001. Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference. Rediscovering Measurement in the Age of Informatics (Cat. No.01CH 37188)*. Vol. 3. May 2001, 1554–1558 vol.3. DOI: 10.1109/IMTC.2001.929465.
- [11] J. Nilsson, A. K. Gupta, and P. Händel. "Foot-mounted inertial navigation made easy". In: *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. Oct. 2014, pp. 24–29. DOI: 10.1109/IPIN.2014.7275464.

-
- [12] John-Olof Nilsson et al. “Cooperative localization by dual foot-mounted inertial sensors and inter-agent ranging”. In: *EURASIP Journal on Advances in Signal Processing* 2013.1 (Oct. 2013), p. 164. ISSN: 1687-6180. DOI: 10.1186/1687-6180-2013-164. URL: <https://doi.org/10.1186/1687-6180-2013-164>.
- [13] K. Papafotis and P. P. Sotiriadis. “MAG.I.C.AL.—A Unified Methodology for Magnetic and Inertial Sensors Calibration and Alignment”. In: *IEEE Sensors Journal* 19.18 (Sept. 2019), pp. 8241–8251. ISSN: 1530-437X. DOI: 10.1109/JSEN.2019.2919179.
- [14] Ross Stirling et al. “An innovative shoe-mounted pedestrian navigation system”. In: Jan. 2003, pp. 110–5.
- [15] Seong Yun Cho and Chan Gook Park. “MEMS Based Pedestrian Navigation System”. In: *Journal of Navigation* 59 (Jan. 2006), pp. 135–153. DOI: 10.1017/S0373463305003486.
- [16] Y. Zhong et al. “A New Drone Accelerometer Calibration Method”. In: *2018 37th Chinese Control Conference (CCC)*. July 2018, pp. 9928–9933. DOI: 10.23919/ChiCC.2018.8482568.

7

MAG.I.NAV.

MAG.I.NAV. is a MAGnetic - Inertial NAVigation algorithm, offering long-term-accurate attitude, velocity and position estimation in the inertial frame. It is implemented in a dedicated pedestrian navigation system using a three-axis accelerometer, a three-axis gyroscope and a three-axis magnetometer mounted on the shoe of a walking person. MAG.I.NAV. compensates for the accumulated attitude errors caused by the gyroscope's measurement error. It does so by using a second attitude estimate derived by combining the accelerometer's and the magnetometer's measurements under zero-velocity and magnetic-disturbance-free conditions. Instead of using a complicated Attitude Heading Reference System (AHRS), MAG.I.NAV. employs the computationally efficient TRIAD algorithm along with a zero-velocity detection and a magnetic-disturbance detection algorithms.

The developed system is tested using commercial, low-cost inertial and magnetic sensors in an outdoor environment. It achieves high long-term accuracy, yielding a position error smaller than 0.25% of the total walking distance in a 20-minute, 1.3km long walk.

7.1 Introduction

Satellite-based navigation systems (GPS, Galileo, GLONASS etc.) are the dominant navigation technology. Different grades of satellite navigation systems provide sufficient accuracy for a variety of applications, ranging from low-cost commercial ones to high-end industrial and military ones. However, even when military grade systems are concerned, all satellite navigation systems have the same inherent disadvantages; they have limited refresh rate, they don't work in indoor environments and they are susceptible to jamming. To overcome these limitations, several alternative navigation technologies have been developed over the past decades.

Inertial navigation systems (INS) use inertial sensors (accelerometers and gyroscopes) to calculate the attitude, velocity and position of a moving object. They were originally developed for rocket guidance during the second world war and since then they are widely used. Inertial navigation systems are nowadays used in many marine, aerospace, military and even commercial applications.

Their wider use was enabled by the development of micro-electro-mechanical (MEM) inertial sensors over the past decades. Due to their miniature size and low cost, MEM inertial sensors are embedded to many commercial devices such as smartphones, activity trackers and alarm systems giving rise to the development of inertial navigation applications.

Pedestrian navigation using inertial sensors has gained significant attention over the past years. Many works use miniature inertial sensors, mounted on a human's body and propose different algorithms to estimate the human's attitude, velocity and position [13, 6, 8, 1, 21, 22, 5, 12, 14, 30, 15, 11, 32, 9, 2]. A crucial design aspect of such systems, which is the main research topic, is how to compensate for the large error characteristics of the inertial sensors [10, 23].

In most pedestrian inertial navigation systems, shoe-mounted inertial sensors and the zero velocity update method are used to correct the attitude, velocity and position estimates [30, 5, 21, 1, 8, 13, 6]. Specifically, they use a zero velocity detection algorithm which detects the stance phase of the human walking, during which the velocity of the shoe is zero. Then, they use this information to estimate the attitude, velocity and position errors, typically by employing a Kalman filter.

The zero velocity update method yields in accurate navigation results for short time periods. However, when low-cost sensors are used, small errors are accumulated with time and in long-term the navigation accuracy is significantly degraded. The greatest part of the position error is due to the attitude error introduced by the gyroscope's noise, offset drift and residual calibration errors [23].

In applications where cost is of no concern, expensive, factory calibrated gyroscopes are used to minimize the attitude error. For commercial applications however, where cost is a critical design aspect, a popular approach is to combine the zero velocity update method with an AHR algorithm typically using the measurements of a three-axis magnetometer to derive an accurate attitude estimate in long-term. Many works propose different AHR algorithms [18, 17, 7, 19, 2, 25, 4, 26] based on estimation [19, 4, 26], optimization [17] or filtering [18, 7] techniques. Despite the fact that existing AHR algorithms provide significant improvement in attitude estimation, they all impose increased computational burden making them difficult to employ in applications where the computational power is limited.

This work introduces MAG.I.NAV., a computationally efficient, long-term accurate pedestrian navigation algorithm using inertial and magnetic field sensors. More specifically, MAG.I.NAV. uses a three-axis accelerometer, a three-axis gyroscope and a three-axis magnetometer mounted on the shoe of a walking person to derive the person's attitude, velocity and position. It is based on the popular zero velocity update method and introduces a computationally efficient attitude estimation scheme using the TRIAD algorithm and both the accelerometer's and the magnetometer's measurements.

To evaluate the performance of MAG.I.NAV. algorithm, low-cost MEM inertial and magnetic field sensors were used. The sensors were calibrated and aligned using MAG.I.CAL. methodology [24] and no special piece of equipment. The performance of MAG.I.NAV. is evaluated in terms of accuracy and computational efficiency in a 20-minute, 1.3km walk in a suburban environment

and compared to the performance of similar algorithms based on two popular and highly cited AHRS. MAG.I.NAV achieves an extremely small position error, below 0.2% of the total walk distance.

7.2 The Proposed Inertial Navigation System

In this section, the proposed pedestrian inertial navigation system is introduced and every functional block is analyzed in detail. The notation used along the paper is first presented.

7.2.1 Notation and Assumptions

Among the different notations used in the literature to describe the basic kinematic quantities (acceleration, velocity, position etc), we follow that of [10] denoting a kinematic quantity, x , as $x_{\beta\alpha}^{\gamma}$, where α is the object frame, β is the reference frame and γ represents the resolving frame.

When the measurement of an inertial or magnetic sensor is concerned, the object frame and the reference frame are fixed and correspond to the sensor's coordinate frame and the inertial frame respectively. Thus, for notation simplicity, a measurement is denoted as y_{γ} .

The frame transformation matrix, which transforms the resolving frame of a kinematic quantity from α to β is denoted as C_{α}^{β} . Furthermore, the frame transformation matrix C_b^i which relates the sensors' coordinate (body) frame, b , to the inertial frame, i , is referred to as attitude.

Given a vector $x = [x_1 \ x_2 \ x_3]^T$, the Cross Product Matrix $[x \times]$ is defined as [27]

$$[x \times] = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

The notation used in the rest of this work is presented in Table 7.1. In addition, we assume that the three sensors are *fixed on the same rigid platform, which is mounted on the shoe of a walking person*. All three sensors are considered to be individually calibrated and their sensitivity axes to be aligned. The three sensors are sampled simultaneously with a common, constant sampling rate, τ_s . Finally, we assume that when the experiment starts, the sensors' platform is still and away from magnetic disturbance.

Coordinate Frames	
b	sensors' platform (body) frame
i	inertial frame
Kinematic Quantities	
$f_{\beta\alpha}^\gamma : \mathbb{R} \rightarrow \mathbb{R}^3$	specific force
$\omega_{\beta\alpha}^\gamma : \mathbb{R} \rightarrow \mathbb{R}^3$	angular velocity
$u_{\beta\alpha}^\gamma : \mathbb{R} \rightarrow \mathbb{R}^3$	velocity
$p_{\beta\alpha}^\gamma : \mathbb{R} \rightarrow \mathbb{R}^3$	position
$C_b^i : \mathbb{R} \rightarrow SO(3)$	attitude
Sensors' Measurements	
$f_\gamma : \mathbb{Z} \rightarrow \mathbb{R}^3$	accelerometer's measurement
$\omega_\gamma : \mathbb{Z} \rightarrow \mathbb{R}^3$	gyroscope's measurement
$m_\gamma : \mathbb{Z} \rightarrow \mathbb{R}^3$	magnetometer's measurement
$\tau_s \in \mathbb{R}$	sensors' sampling period
System's Outputs	
$u : \mathbb{Z} \rightarrow \mathbb{R}^3$	velocity output
$p : \mathbb{Z} \rightarrow \mathbb{R}^3$	position output
$C : \mathbb{Z} \rightarrow SO(3)$	attitude output
Other Notation	
$I_n \in SO(3)$	$n \times n$ identity matrix
$O_n \in \mathbb{R}^3$	$n \times n$ matrix of zeros
$\ \cdot \ $	Euclidean norm

Table 7.1: Notation

7.2.2 Top-Level System Architecture

The architecture of the proposed inertial navigation system is presented in Figure 7.1. It uses the measurements of a three-axis accelerometer, f_b , a three-axis gyroscope, ω_b , and a three-axis magnetometer, m_b , to estimate the attitude, C , the velocity, v , and the position, p , of a walking person.

For every sensors' measurement, the system derives a first estimate of the attitude, C_K , using the measurement of the gyroscope. In order to correct the long-term accumulated attitude error, the proposed system uses a second, independent of the first one, estimate of the attitude, derived using the measurements of the accelerometer and the magnetometer. To do so, when a zero velocity condition is detected ($ZV = 1$ in Figure 7.1) and if there is no magnetic disturbance ($D = 1$ in Figure 7.1) it applies the TRIAD algorithm to the measurements of the two sensors to derive the second attitude estimate denoted as C_{FM} and update the attitude estimate C_{GFM} accordingly as shown in Figure 7.1.

The next step is to derive a first velocity, v_K , and position, p_K , estimate. This is done by using

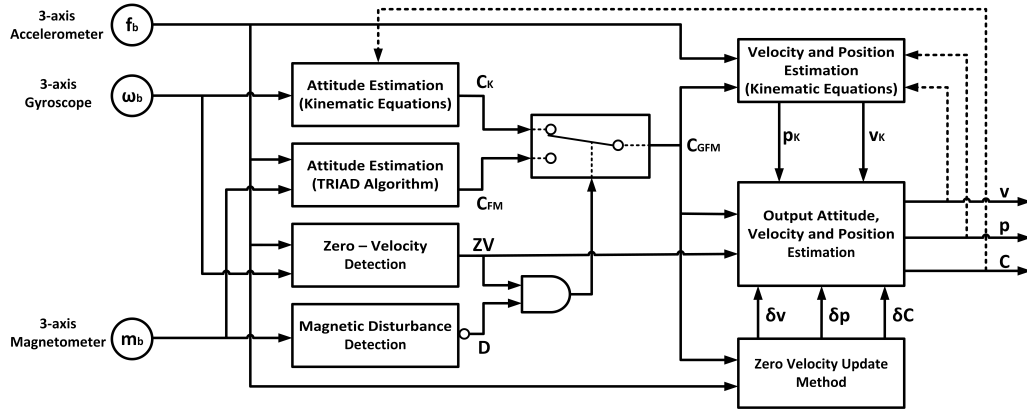


Figure 7.1: Architecture of the proposed pedestrian inertial navigation system.

the corrected attitude, C_{GFM} , along with the accelerometer’s measurement and the kinematic equations.

The zero velocity update method is finally employed to further improve the attitude, velocity and position estimates. It is realized using a Kalman filter and it is applied every time a zero velocity condition is detected ($ZV = 1$ in Figure 7.1).

7.2.3 Zero Velocity Detection

The zero velocity detection algorithm is an important component of pedestrian navigation systems using shoe-mounted sensors as part of the widely used zero velocity update method. In the proposed system, the zero velocity information is used both to implement the zero velocity update method (see Section 7.2.9) and to derive a long-term accurate attitude estimate using the magnetometer’s and the accelerometer’s measurements (see Section 7.2.6). Zero velocity detection algorithms use the inertial sensors’ data along with information about the human walking pattern to detect the stance phase of the walking [29, 31, 5, 6]. More advanced algorithms exploit more sensors to improve the zero velocity detection accuracy (e.g. pressure sensors [1]). Several review papers compare the performance characteristics of existing zero velocity detection algorithms [29, 6]. In practice, for navigation purposes, different zero velocity detection algorithms tend to have similar performance [6].

In this work, we use the measurements of the accelerometer, f_b , and the gyroscope, ω_b , to detect when the velocity of the sensors’ platform is zero based on the algorithm introduced in [29]. We begin by calculating the mean value of the accelerometer’s measurements using a rolling window of size N , as follows

$$f_b(\bar{k}) = \frac{1}{N} \sum_k^{(k+N-1)} f_b(k) \quad (7.1)$$

Then, to decide if a zero velocity condition is present at time k , we calculate the quantity

$$T(k) = \sum_k^{(k+N-1)} \left(\frac{1}{\sigma_a^2} \left\| f_b(k) - g \frac{\bar{f}_b(k)}{\|\bar{f}_b(k)\|} \right\|^2 + \frac{1}{\sigma_g^2} \|\omega_b(k)\|^2 \right) \quad (7.2)$$

where σ_a^2 and σ_g^2 are the noise variances of the accelerometer and the gyroscope respectively and g is the gravitational acceleration. According to [29], we define

$$ZV(k) = \begin{cases} 1, & T(k) < \gamma_z \\ 0, & otherwise \end{cases} \quad (7.3)$$

where γ_z is an appropriate threshold parameter. In (7.3), $ZV(k) = 1$ indicates the existence of a zero velocity condition.

Special care should be given to the parameterization of the zero velocity detection algorithm. The algorithm requires knowledge of the accelerometer's and gyroscope's error variances. In many cases, when low cost sensors are used, the error characteristics of the sensors are not announced by their manufacturer and must be experimentally derived. In addition the threshold value used by the algorithm to decide about the zero velocity condition must be carefully selected and fine tuned to achieve high performance.

7.2.4 Magnetic Disturbance Detection

Attitude determination using accelerometer's and magnetometer's measurements via the TRIAD algorithm, requires that the two sensors are still and away from magnetic disturbance. The stillness condition is ensured during the stance phase of the human walking, when the shoe is touching the ground. Magnetic disturbances on the other hand are more complex to identify as they distort the earth's magnetic field in a non-predictable way.

In this work we use the magnitude of the measured magnetic field to detect magnetic disturbance. To do so, we exploit the fact that when no magnetic disturbance is present, the magnitude of the measured magnetic field, m_b , is equal to that of the earth's magnetic field, m_{earth} . Assuming that at the beginning of the experiment the magnetometer is placed away from magnetic disturbance, and taking into account the sensor's noise and errors, we write

$$\|m_{earth}\| \approx \|m_b(0)\| \quad (7.4)$$

We consider and use a tolerance parameter m_{thr} to detect whether there is or there is not magnetic disturbance. To this purpose, we define

$$D(k) = \begin{cases} 0, & \left| \|m_{earth}\| - \|m_b(k)\| \right| > m_{thr} \\ 1, & otherwise \end{cases} \quad (7.5)$$

When the difference is above the threshold value, m_{thr} , it is $D(k) = 0$ indicating the existence of magnetic disturbance. In contrast when no magnetic disturbance is present, it is $D(k) = 1$.

Note that in some rare cases the magnetic disturbance could rotate the earth's magnetic field without altering its magnitude, a case which is not detected by (7.5) due to the rotational invariance of the Euclidean norm. However, in pedestrian navigation applications, the orientation is frequently updated and thus a false positive or false negative result of (7.5) does not degrade the navigation accuracy in long-term.

7.2.5 Attitude Estimation Using Gyroscope's Measurements

The standard kinematic equations are used to derive a first estimate of the sensors' platform attitude. We begin by calculating the attitude at time t , $C_b^i(t) \in SO(3)$. To this end, we write [10]

$$\dot{C}_b^i(t) = C_b^i(t)[\omega_{ib}^b(t) \times] \quad (7.6)$$

Given (7.6) and assuming that the angular velocity of the platform is constant during the short time period between consecutive samplings, $[t, t + \tau_s]$, it is

$$C_b^i(t + \tau_s) = C_b^i(t) \exp([\omega_{ib}^b(t) \times] \tau_s) \quad (7.7)$$

Based on (7.7), the attitude at time $k\tau_s$ using the measurement of the gyroscope, $\omega_b(k)$, is approximated by

$$C_b^i(k\tau_s) \approx C_b^i((k-1)\tau_s) (I_3 + [\omega_b(k) \times] \tau_s) \quad (7.8)$$

Equation (7.8) implies that the estimated attitude at time $k\tau_s$, $C_b^i(k\tau_s)$, is affected from the accumulated noise of all the previous gyroscope's measurements. In the proposed system, we use the output attitude, $C(k-1)$, as feedback to reset the accumulated error when a more accurate attitude estimate is available (either from the zero velocity update method or from the TRIAD algorithm). Thus, according to (7.8), we define

$$C_K(k) \triangleq C(k-1) (I_3 + [\omega_b(k) \times] \tau_s) \quad (7.9)$$

To derive (7.9), the power series of the matrix exponent of (7.7) is truncated to the first order. This approximation may introduce significant error when the sampling period of the gyroscope, τ_s , is not sufficiently small. The accuracy of (7.9) can be increased in exchange for computational resources by using a higher order approximation of (7.7) or a higher gyroscope's output rate.

7.2.6 Attitude Estimation via TRIAD Algorithm Using Accelerometer's and Magnetometer's Measurements

In inertial navigation systems attitude is typically calculated using gyroscope's measurements and equation (7.9) or a higher order approximation of (7.7). This approach however, results in the accumulation of significant attitude error in long-term caused by the gyroscope's noise, offset

drift and residual calibration errors. In this section, we exploit the TRIAD algorithm [3, 28] to calculate a second estimate of the sensor's platform attitude using the measurements of the accelerometer and the magnetometer.

The TRIAD algorithm works as follows. Let a_1, a_2, b_1 and b_2 be four 3×1 unit vectors and R be a rotation matrix in $SO(3)$ such that $a_2 = Ra_1$ and $b_2 = Rb_1$. TRIAD takes as inputs the four vectors and derives R . A detailed description of the TRIAD algorithm is presented in Appendix ?? . It is convenient to consider the TRIAD algorithm as a function of the four vectors, i.e.

$$R = \text{TRIAD}(a_1, b_1, a_2, b_2) \quad (7.10)$$

Now, consider the the measurements of the accelerometer, $f_b(k_c)$, and the magnetometer, $m_b(k_c)$, at discrete time $k = k_c > 0$. Assume that at time k_c the sensors' platform is still ($ZV = 1$ in Figure 7.1) and that there is no magnetic disturbance ($D = 1$ in Figure 7.1). Using the measurements of the two sensors and the TRIAD algorithm we derive the following attitude estimate

$$C_{FM}(k_c) = C(0) \text{ TRIAD}(f_b(0), m_b(0), f_b(k_c), m_b(k_c)) \quad (7.11)$$

where $C(0)$, $f_b(0)$ and $m_b(0)$ are captured at the beginning of the experiment, while the sensors' platform is still and away from magnetic disturbance.

7.2.7 Output Attitude Derivation

The output attitude, C , is derived by combining the attitude estimates C_K and C_{FM} , as well as the attitude correction term, δC which is derived by the zero velocity update method (see Section 7.2.9). The operation of the proposed system, is divided in three states as shown in Table 7.2 , according to the outputs of the zero velocity detection algorithm, ZV , and the magnetic disturbance detection algorithm, D .

	$D = 0$	$D = 1$
$ZV = 0$	State 1	
$ZV = 1$	State 2	State 3

Table 7.2: Operation states.

During State 1, the attitude is calculated using the kinematics and equation (7.9). In State 2, the zero velocity update method is used to provide a small attitude correction, compensating for the gyroscope's errors. Finally, during State 3, the accelerometer's and magnetometer's measurements are used to derive an accurate attitude estimate via TRIAD algorithm. Summarizing, the output attitude estimate, C , during each operation state is derived as follows:

$$C(k) = \begin{cases} C_K(k), & \text{State1} \\ \delta C C_K(k), & \text{State2} \\ C_{FM}, & \text{State3} \end{cases} \quad (7.12)$$

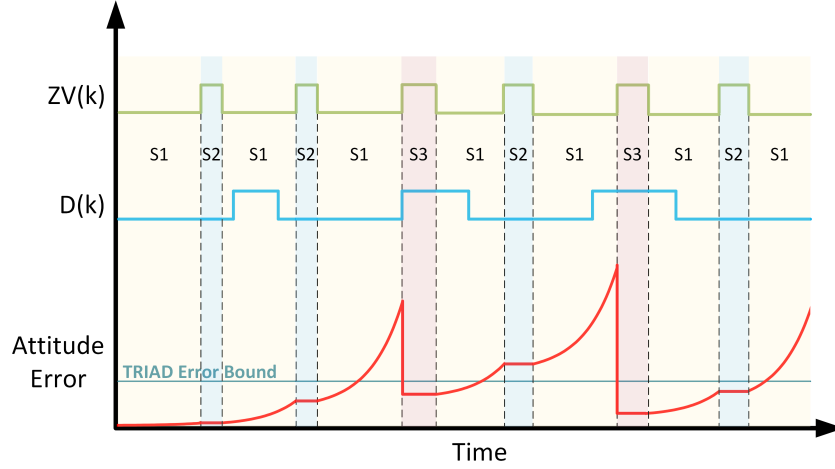


Figure 7.2: Qualitative representation of the attitude error when using the proposed attitude estimation scheme. The three operation states are denoted as S1, S2 and S3 respectively.

A qualitative representation of the output attitude error when using the introduced attitude estimation scheme of (7.12) is presented in Figure 7.2. As seen in Figure 7.2, during State 1, the attitude error, mainly caused by the gyroscope's drift, is accumulated and grows exponentially in time. During State 2, the zero velocity update method (see Section 7.2.9) provides a small attitude correction and prevents the error from rising. However, this small correction cannot compensate for the accumulated attitude error which eventually rises significantly over time (this is also demonstrated using real sensors' measurements in Section 7.3). The attitude error during State 3 depends only on the error of the sensors' measurements at a particular time k and it is bounded¹. Thus, during State 3, an accurate, independent of the previous ones, attitude estimate is derived. In Figure 7.2, it is demonstrated that if such accurate attitude estimates are frequently available, the attitude error does not rise significantly over time, leading to a long-term accurate attitude estimation.

7.2.8 Velocity and Position Estimation Via Kinematic Equations Using Accelerometer's Measurements

In order to derive a first estimate of the velocity and the position, we define the intermediate attitude C_{GFM} following Figure 7.1 as

$$C_{GFM}(k) = \begin{cases} C_{FM}(k), & ZV = 1 \text{ and } D = 1 \\ C_K(k), & \text{otherwise} \end{cases} \quad (7.13)$$

¹Here we assume that the attitude error introduced by the measurement error of $C(0)$, $f_b(0)$ and $m_b(0)$ is small and thus it is neglected. This is a rational assumption, because multiple measurements and averaging can be used to accurately determine $C(0)$, $f_b(0)$ and $m_b(0)$.

We transform the accelerometer's measurement (which is naturally expressed in the sensors' platform frame) to the inertial frame using $C_{GFM}(k)$ as follows

$$f_i(k) = C_{GFM}(k)f_b(k) \quad (7.14)$$

Then, in order to cancel the effect of the gravity acceleration to the accelerometer measurement, we define

$$a(k) = f_i(k) + g_i \quad (7.15)$$

where g_i is the 3×1 gravitational acceleration vector expressed in the inertial frame. The discrete-time velocity and position vectors are updated as

$$v_K(k) = v_K(k-1) + a(k)\tau_s \quad (7.16)$$

and

$$p_K(k) = p_K(k-1) + \frac{1}{2}(v_K(k) + v_K(k-1))\tau_s \quad (7.17)$$

7.2.9 Zero Velocity Update Method

The zero velocity update method is realized by an error-state Kalman filter similar to [10]. The filter uses the measurements of the accelerometer and the gyroscope to estimate the error of the attitude, the velocity and the position as well as the offset vectors of the two sensors.

The author in [10] uses the following 15×1 Kalman filter's state vector

$$x_{zv} = [\delta\psi^T \quad \delta v^T \quad \delta p^T \quad b_a^T \quad b_g^T]^T \quad (7.18)$$

where $\delta\psi$ is the 3×1 attitude error expressed in Euler angles, δv is the 3×1 velocity error, δr is the 3×1 position error, b_a is the 3×1 accelerometer's offset vector and b_g is the 3×1 gyroscope's offset vector. Using the kinematic equations and assuming a small error in the measurements of the accelerometer and the gyroscope, we derive the following state propagation model for the Kalman filter similar to [10]

$$x_{zv}(k+1) = \Phi_{zv}(k)x_{zv}(k) + w_{zv}(k) \quad (7.19)$$

where w_{zv} is assumed to be a white noise sequence and

$$\Phi_{zv}(k) = \begin{bmatrix} I_3 & 0_3 & 0_3 & 0_3 & C_{\tau_s}(k) \\ F(k) & I_3 & 0_3 & C_{\tau_s}(k) & 0_3 \\ 0_3 & I_3\tau_s & I_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & I_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & I_3 \end{bmatrix} \quad (7.20)$$

where

$$F(k) = [-f_i(k) \times] \tau_s$$

and

$$C_{\tau_s}(k) = C_{GFM}(k) \tau_s$$

When a zero velocity condition is present, the velocity output of the kinematic equations (see Section 7.2.8) should be ideally zero. However, due to the sensors' non-idealities (noise, residual calibration error, etc.), in practice, this is not the case; kinematic equations give a small (error) velocity output even when the sensors' platform is still. This velocity is used as a measurement of the velocity error, δv , in the Kalman filter, implying the following form of the filter's measurement equation

$$z_{zv}(k) = u_K(k) = H_{zv} x_{zv}(k) + v_{zv}(k) \quad (7.21)$$

where v_{zv} is a white noise sequence and

$$H_{zv} = \begin{bmatrix} 0_3 & I_3 & 0_3 & 0_3 & 0_3 \end{bmatrix}^T \quad (7.22)$$

The initialization of the diagonal covariance matrices Q_{zv} and R_{zv} is crucial to achieve high performance. Let σ_a^2 and σ_g^2 be the variance of the accelerometer's and the gyroscope's noise respectively ². Also, let σ_{ba}^2 and σ_{bg}^2 be the variances of their offset as it drifts in time. Using them we form Q_{zv} as follows,

$$Q_{zv} = \text{diag} \left([\sigma_g^2 1_{1 \times 3} \quad \sigma_a^2 1_{1 \times 3} \quad 0_{1 \times 3} \quad \sigma_{ba}^2 1_{1 \times 3} \quad \sigma_{bg}^2 1_{1 \times 3}] \right) \tau_s \quad (7.23)$$

The measurement covariance matrix, R_{zv} , is similarly formed by assuming the variance of the velocity measurement when a zero-velocity condition is present, σ_v^2

$$R_{zv} = \sigma_v^2 I_3 \quad (7.24)$$

The variances used to form R_{zv} and Q_{zv} in (7.23) and (7.24) cannot be typically derived from the sensors' documentation, especially in the case of low cost, MEM sensors. Thus, both R_{zv} and Q_{zv} are usually derived and fine tuned experimentally.

Note that the derived attitude correction term, $\delta\psi$, is expressed in Euler angles. Using small angle approximation, the corresponding rotation matrix is defined as

$$\delta C = [\delta\psi \times] \quad (7.25)$$

²Here, without loss of generality, we assumed that the three axes of each sensor present similar noise characteristics.

7.2.10 System's Initialization

The proposed system derives the navigation solution in the inertial frame. Thus, the position vector is initialized as $p(0) = [0 \ 0 \ 0]^T$. Under the assumption that at time $t = 0$, the sensors' platform is still, the velocity vector is initialized as $v(0) = [0 \ 0 \ 0]^T$.

Deriving the initial attitude, i.e. the rotation between the inertial frame and the sensors' platform frame at time $t = 0$, is not a trivial task. However, aligning the z axes between the two coordinate frames is crucial in order to be able to subtract the gravitational acceleration when solving the kinematic equations (see equation (7.15)). To do so, we consider the gravitational acceleration vector in the inertial frame, $g_{ib}^i = [0 \ 0 \ g]^T$, and the accelerometer's measurement at $t = 0$ while the sensor is still, $f_b(0)$.³ Then we use Rodrigues' rotation formula [20] to derive the rotation matrix $C(0)$ which aligns the z -axes of the two coordinate frames.

7.2.11 Algorithmic Implementation

The algorithmic implementation of the proposed system including the Kalman filter's recursive equations is presented in Algorithm 14.

³Gravitational acceleration, g , depends on the location of the experiment and can be found in relevant models. However in most cases it is sufficient to assume that it is constant and equal to 9.80665 m/s^2 .

Algorithm 14: Proposed Navigation Algorithm.

System's Initialization

The system initialization is done while the sensors' platform is still and away from magnetic disturbances

Step 1: Calculate the initial attitude, $C(0)$, according to Section 7.2.10.

Step 2: Capture the accelerometer's and magnetometer's measurements $f_b(0)$ and $m_b(0)$ respectively.

Step 3: Initialize velocity and position vectors according to Section 7.2.10.

Kalman Filter's Initialization

Step 4: Form the noise covariance matrices, Q_{zv} and R_{zv} according to (7.23) and (7.24) respectively.

Step 5: Form the measurement matrix H_{zv} following (7.22).

Step 6: Initialize the error covariance matrix $P_{zv}(0) = O_{15}$.

Navigation Equations

The navigation equations (steps 7 - 15) are evaluated for every sensors' sample, $k = 1, 2, \dots, K$.

Step 7: Calculate $C_K(k)$ using (7.9).

Step 8: Calculate $ZV(k)$ and $D(k)$ according to (7.3) and (7.5) respectively.

Step 9: If $ZV(k) = 1$ and $D(k) = 1$, calculate $C_{FM}(k)$ according to (7.11).

Step 10: Set $C_{GFM}(k)$ following (7.13).

Step 11: Calculate $u_K(k)$ and $p_K(k)$ following (7.16) and (7.17).

Step 12: Form the Kalman filter's state transition matrix, $\Phi_{zv}(k)$ using (??).

Step 13: Calculate the error covariance matrix

$$P_{zv}^-(k) = \Phi_{zv}(k)P_{zv}(k-1)\Phi_{zv}(k)^T + Q_{zv}.$$

Step 14: If $ZV(k) = 1$:

14.1) Calculate Kalman gain $K_{zv}(k) = P_{zv}(k)H_{zv}^T(H_{zv}P_{zv}(k)H_{zv}^T + R_{zv})^{-1}$.

14.2) Calculate $x_{zv}(k) = K_{zv}(k)v_K(k)$.

14.3) Update the error covariance matrix $P_{zv}(k) = (I_{15} - K_{zv}(k)H_{zv})P_{zv}^-(k)$.

14.4) Extract $\delta\psi(k)$, $\delta v(k)$ and $\delta r(k)$ from $x_{zv}(k)$

14.5) Calculate the system's outputs, $C(k) = [\delta\psi(k) \times]C_{GFM}(k)$, $v(k) = v_K(k) - \delta v(k)$ and $p(k) = p_K(k) - \delta p(k)$.

14.6) Extract $b_a(k)$ and $b_g(k)$ from $x_{zv}(k)$ and subtract them from the next $(k+1)$ measurement of the accelerometer and the gyroscope respectively.

Step 15: If $ZV(k) = 0$: $C(k) = C_{GFM}(k)$, $v(k) = v_K(k)$ and $p(k) = p_K(k)$.

7.3 Experimental Results

In order to evaluate the proposed inertial navigation algorithm, we used an inertial measurement unit (IMU) based on the low-cost LSM9DS1 system in package (SiP) by STMicroelectronics. It contains a three-axis accelerometer, a three-axis gyroscope and a three-axis magnetometer, the most important performance specifications of which are presented Table 7.3.

The proposed algorithm was tested during an 20 minute and about 1.3km long walk around the campus of the National Technical University of Athens, Greece. The walk path is shown in Figure 7.3 along with some important landmarks. As seen in in Figure 7.3, along the walk path there are several sources of magnetic disturbance such as buildings and parking areas.

Specification	Value
Measurement Range (A)	$\pm 16g$
Measurement Range (G)	$\pm 2000^\circ/s$
Measurement Range (M)	$\pm 4Gauss$
Sampling Rate (A, G)	238Hz
Sampling Rate (M)	80Hz
Resolution (A, G, M)	16Bits

Table 7.3: Basic performance characteristics of the accelerometer (A), gyroscope (G) and magnetometer (M) included in the LSM9DS1 SiP.

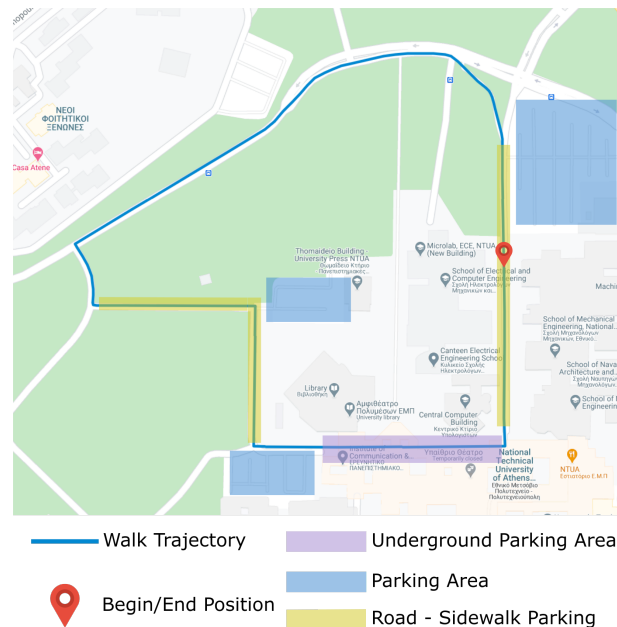


Figure 7.3: Walk path inside the campus of the National Technical University of Athens, Greece.

First, we demonstrate the long-term accuracy of the proposed system compared to existing pedestrian navigation systems using the zero velocity update method alone. To do so, we reconstructed the walk path using only the accelerometer's and gyroscope's measurements and

the zero velocity update method. As seen in Figure 7.4, the calculated attitude drifts after a few meters and the reconstructed walk path diverges significantly from the reference one.

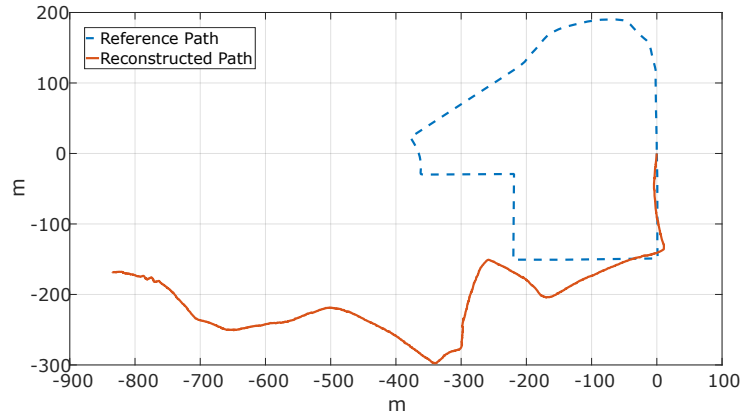


Figure 7.4: Reconstructed walk path using accelerometer's and gyroscope's measurements and the zero velocity update method.

The reconstructed walk path using all three sensors' measurements and the proposed algorithm is presented in Figure 7.5. Observe that the reconstructed path in Figure 7.5 successfully tracks the reference one in long-term.

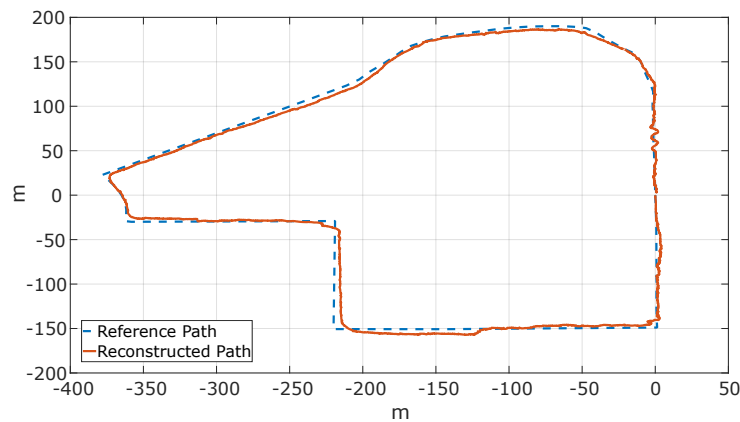


Figure 7.5: Reconstructed walk path using the measurements of both inertial and magnetic sensors, the zero velocity update method and the proposed attitude correction scheme.

In Figure 7.6 some accuracy characteristics along the reconstructed path are highlighted. 1) The actual path formed a close-loop, i.e. it started and ended at exactly the same point. As seen in Figure 7.6, the reconstructed path presents an error of $3.08m$ from start point to end point. This is a rather impressive performance as the error is below 0.25% of the total walk distance. 2) The high refresh rate and measurement accuracy of the proposed system allows for the identification of walking patterns. This is demonstrated in Figure 7.6 where a road crossing and a zig-zag pattern walking are easily identified. 3) A closer inspection of the reconstructed trajectory reveals the position correction introduced by both the introduced attitude correction scheme and zero velocity update method in three consecutive steps.

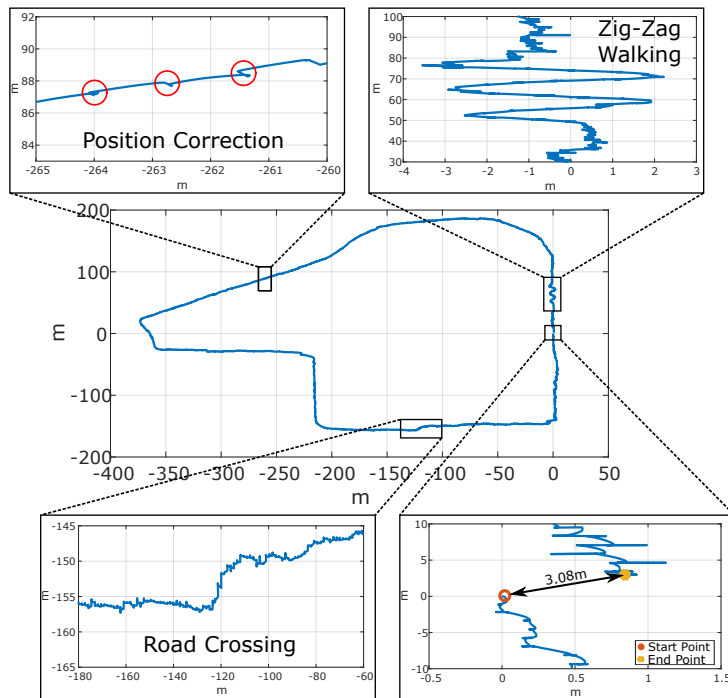


Figure 7.6: Reconstructed walk path with highlighted some important accuracy characteristics.

The proposed algorithm is compared to existing pedestrian navigation algorithms in terms of accuracy and computational efficiency using two popular, highly cited AHR algorithms. The zero-velocity update method as realized in [10] is combined with the Madgwick's [17] and Mahony's [18] attitude estimation algorithms similar to [16]. The same measurement set along the walking path of Figure 7.3 is used to reconstruct the walking path using the three algorithms.

In Figure 7.7 the three algorithms are compared in terms of accuracy and long-term stability. The proposed algorithm outperforms both the Madgwick's AHRS and Mahony's AHRS based ones as it successfully tracks the reference path through the whole walking distance. The algorithm based on Madgwick's AHRS accurately tracks the reference path and slightly drifts only during the last 250m while the Mahony's AHRS based algorithm seems to be less resilient to the magnetic disturbances along the walking path.

In Table 7.4 the three algorithms are compared in terms of computational efficiency. The three algorithms were executed in MATLAB running on a typical quad-core, 8GB RAM PC. Note that as all three algorithms are based on the same Kalman filter framework to implement the zero velocity update method. Thus, the attitude estimation algorithm is their only substantial difference as far as the computational burden of each algorithm is concerned. To ensure a more fair comparison, each algorithm was executed five times using the same dataset and the mean execution time of all algorithms is presented in Table 7.4.

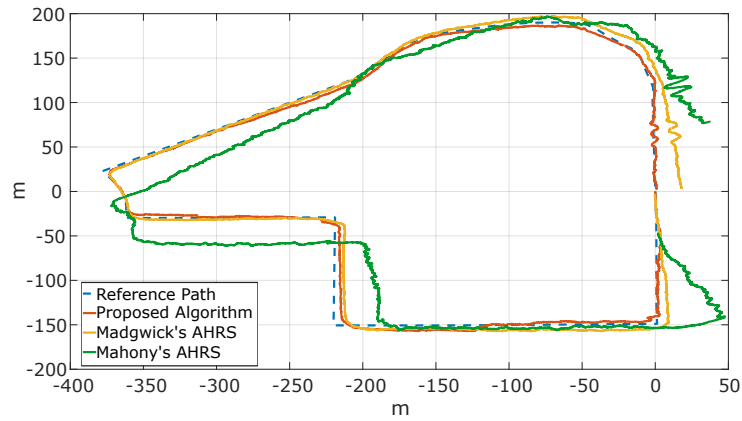


Figure 7.7: Reconstructed walk path using the measurements of both inertial and magnetic sensors, the zero velocity update method and the proposed attitude correction scheme.

Algorithm	Mean Execution Time
Proposed	10.67s
[10] + Madgwick's AHRS[17]	19.96s
[10] + Mahony's AHRS[18]	18.32

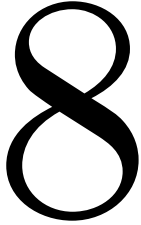
Table 7.4: Mean execution time of the proposed algorithm compared to the mean execution time of the algorithms based on Madgwick's AHRS and Mahony's AHRS.

References

- [1] O. Bebek et al. “Personal Navigation via High-Resolution Gait-Corrected Inertial Measurement Units”. In: *IEEE Transactions on Instrumentation and Measurement* 59.11 (Nov. 2010), pp. 3018–3027. ISSN: 0018-9456. DOI: 10.1109/TIM.2010.2046595.
- [2] J. Bird and D. Arden. “Indoor navigation with foot-mounted strapdown inertial navigation and magnetic sensors [Emerging Opportunities for Localization and Tracking]”. In: *IEEE Wireless Communications* 18.2 (2011), pp. 28–35. DOI: 10.1109/MWC.2011.5751293.
- [3] H. D. Black. “A passive system for determining the attitude of a satellite”. In: *AIAA Journal* 2.7 (July 1964), pp. 1350–1351. ISSN: 0001-1452. DOI: 10.2514/3.2555. URL: <https://doi.org/10.2514/3.2555>.
- [4] D. Choukroun, I.Y. Bar-Itzhack, and Y. Oshman. “Novel quaternion Kalman filter”. In: *IEEE Transactions on Aerospace and Electronic Systems* 42.1 (2006), pp. 174–190. DOI: 10.1109/TAES.2006.1603413.
- [5] Raul Feliz, Eduardo Zalama, and Jaime Gómez-García-Bermejo. “Pedestrian Tracking Using Inertial Sensors”. In: *Journal of Physical Agents* 3 (Jan. 2009). DOI: 10.14198/JoPha.2009.3.1.05.
- [6] C. Fischer, P. Talkad Sukumar, and M. Hazas. “Tutorial: Implementing a Pedestrian Tracker Using Inertial Sensors”. In: *IEEE Pervasive Computing* 12.2 (Apr. 2013), pp. 17–27. ISSN: 1536-1268. DOI: 10.1109/MPRV.2012.16.
- [7] Hassen Fourati. “Heterogeneous Data Fusion Algorithm for Pedestrian Navigation via Foot-Mounted Inertial Measurement Unit and Complementary Filter”. In: *IEEE Transactions on Instrumentation and Measurement* 64.1 (2015), pp. 221–229. DOI: 10.1109/TIM.2014.2335912.
- [8] E. Foxlin. “Pedestrian tracking with shoe-mounted inertial sensors”. In: *IEEE Computer Graphics and Applications* 25.6 (Nov. 2005), pp. 38–46. ISSN: 0272-1716. DOI: 10.1109/MCG.2005.140.
- [9] Paul Groves et al. “Inertial Navigation Versus Pedestrian Dead Reckoning: Optimizing the Integration”. In: vol. 2. Sept. 2007.
- [10] Paul D. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2013.
- [11] Hang Guo et al. “Indoor positioning based on foot-mounted IMU”. In: *Bulletin of the Polish Academy of Sciences Technical Sciences* 63 (Sept. 2015). DOI: 10.1515/bpasts-2015-0074.
- [12] A. R. Jimenez et al. “A comparison of Pedestrian Dead-Reckoning algorithms using a low-cost MEMS IMU”. In: *2009 IEEE International Symposium on Intelligent Signal Processing*. Aug. 2009, pp. 37–42. DOI: 10.1109/WISP.2009.5286542.

- [13] A. R. Jiménez et al. “Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU”. In: *2010 7th Workshop on Positioning, Navigation and Communication*. Mar. 2010, pp. 135–143. DOI: 10.1109/WPNC.2010.5649300.
- [14] R. Jirawimut et al. “A method for dead reckoning parameter correction in pedestrian navigation system”. In: *IMTC 2001. Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference. Rediscovering Measurement in the Age of Informatics (Cat. No.01CH 37188)*. Vol. 3. May 2001, 1554–1558 vol.3. DOI: 10.1109/IMTC.2001.929465.
- [15] Lei Fang et al. “Design of a wireless assisted pedestrian dead reckoning system - the NavMote experience”. In: *IEEE Transactions on Instrumentation and Measurement* 54.6 (Dec. 2005), pp. 2342–2358. ISSN: 0018-9456. DOI: 10.1109/TIM.2005.858557.
- [16] Xin Li and Yang Wang. “Evaluation of AHRS algorithms for Foot-Mounted Inertial-based Indoor Navigation Systems”. In: *Open Geosciences* 11.1 (2019), pp. 48–63. DOI: doi: 10.1515/geo-2019-0005. URL: <https://doi.org/10.1515/geo-2019-0005>.
- [17] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. “Estimation of IMU and MARG orientation using a gradient descent algorithm”. In: *2011 IEEE International Conference on Rehabilitation Robotics*. June 2011, pp. 1–7. DOI: 10.1109/ICORR.2011.5975346.
- [18] Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. “Nonlinear Complementary Filters on the Special Orthogonal Group”. In: *IEEE Transactions on Automatic Control* 53.5 (2008), pp. 1203–1218. DOI: 10.1109/TAC.2008.923738.
- [19] Philippe Martin and Erwan Salaün. “Design and implementation of a low-cost observer-based attitude and heading reference system”. In: *Control Engineering Practice* 18.7 (2010). Special Issue on Aerial Robotics, pp. 712–722. ISSN: 0967-0661. DOI: <https://doi.org/10.1016/j.conengprac.2010.01.012>. URL: <https://www.sciencedirect.com/science/article/pii/S0967066110000201>.
- [20] Richard M. Murray, S. Shankar Sastry Zexiang Li, and S. Shankara Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
- [21] J. Nilsson, A. K. Gupta, and P. Händel. “Foot-mounted inertial navigation made easy”. In: *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. Oct. 2014, pp. 24–29. DOI: 10.1109/IPIN.2014.7275464.
- [22] John-Olof Nilsson et al. “Cooperative localization by dual foot-mounted inertial sensors and inter-agent ranging”. In: *EURASIP Journal on Advances in Signal Processing* 2013.1 (Oct. 2013), p. 164. ISSN: 1687-6180. DOI: 10.1186/1687-6180-2013-164. URL: <https://doi.org/10.1186/1687-6180-2013-164>.

- [23] K. Papafotis and P. P. Sotiriadis. “Exploring the Importance of Sensors’ Calibration in Inertial Navigation Systems”. In: *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2020, pp. 1–4. DOI: 10.1109/ISCAS45731.2020.9181212.
- [24] K. Papafotis and P. P. Sotiriadis. “MAG.I.C.AL.—A Unified Methodology for Magnetic and Inertial Sensors Calibration and Alignment”. In: *IEEE Sensors Journal* 19.18 (Sept. 2019), pp. 8241–8251. ISSN: 1530-437X. DOI: 10.1109/JSEN.2019.2919179.
- [25] V. Renaudin, M. H. Afzal, and G. Lachapelle. “New method for magnetometers based orientation estimation”. In: *IEEE/ION Position, Location and Navigation Symposium*. May 2010, pp. 348–356. DOI: 10.1109/PLANS.2010.5507301.
- [26] Valérie Renaudin and Christophe Combettes. “Magnetic, Acceleration Fields and Gyroscope Quaternion (MAGYQ)-Based Attitude Estimation with Smartphone Sensors for Indoor Pedestrian Navigation”. In: *Sensors* 14.12 (2014), pp. 22864–22890. ISSN: 1424-8220. DOI: 10.3390/s141222864. URL: <https://www.mdpi.com/1424-8220/14/12/22864>.
- [27] Paul G Savage. *Strapdown Analytics, Part 2*. Strapdown Associates, 2007.
- [28] M. D. Shuster and S. D. Oh. “Three-axis attitude determination from vector observations”. In: *Journal of Guidance, Control, and Dynamics* 4.1 (Jan. 1981), pp. 70–77. ISSN: 0731-5090. DOI: 10.2514/3.19717. URL: <https://doi.org/10.2514/3.19717>.
- [29] Isaac Skog et al. “Zero-Velocity Detection - An Algorithm Evaluation”. In: *IEEE Transactions on Biomedical Engineering* 57 (11 2010), pp. 2657–2666. ISSN: 1558-2531. DOI: 10.1109/TBME.2010.2060723.
- [30] Ross Stirling et al. “An innovative shoe-mounted pedestrian navigation system”. In: Jan. 2003, pp. 110–5.
- [31] Zhelong Wang et al. “Stance-Phase Detection for ZUPT-Aided Foot-Mounted Pedestrian Navigation System”. In: *IEEE/ASME Transactions on Mechatronics* 20 (6 2015), pp. 3170–3181. ISSN: 1941-014X. DOI: 10.1109/TMECH.2015.2430357.
- [32] Seong Yun Cho and Chan Gook Park. “MEMS Based Pedestrian Navigation System”. In: *Journal of Navigation* 59 (Jan. 2006), pp. 135–153. DOI: 10.1017/S0373463305003486.



A Class of Low-Noise Inertial Measurement Units

An Inertial Measurement Unit (IMU) architecture estimating angular velocity is introduced. It expands the concept of multi-accelerometers gyroscope-free IMUs (GF-IMUs), taking advantage of their excellent noise performance and resolving their inability to compensate for accelerometers' bias and their requirement for specific accelerometers placement, which makes GF-IMUs inappropriate for real world applications. It does so by embedding the accelerometers in a low-bandwidth closed-loop configuration with a three-axis gyroscope and by addressing the complete non-linear dynamics. Extensive theoretical analysis provides a complete framework for designing low-noise IMUs including sufficient stability criteria for the non-linear system's dynamics as well as output noise models. Simulation results support the theoretical analysis and indicate that even a minimal system using the proposed architecture outperforms the gyroscope in noise performance providing more than 15dB improvement.

8.1 Introduction

Inertial sensors (accelerometers and gyroscopes) are nowadays embedded in several commercial devices such as smartphones, activity trackers, alarm systems and others while they are also used in many high-end, industrial, marine, aerospace and military applications. The fast development of Micro-Electro-Mechanical (MEM) inertial sensors over the past decades enabled the wider use of inertial sensors. Their miniature size and extremely low cost make MEM inertial sensors the ideal choice for a plethora of applications. However, their large error characteristics and measurement noise [11] forbid their use in applications where measurement accuracy is important.

The greatest part of the measurement error of an inertial sensor is static and caused by imperfections of the mechanical and electronic structures of the sensor. The static error is most commonly modeled as a linear combination of different error terms (bias, non-orthogonality, cross-axis sensitivity etc.) and can be compensated by using a proper calibration techniques [21, 14, 28, 29, 7].

On the other hand, dynamic measurement errors such as bias drift and non-deterministic noise are more complicated problems which are most commonly dealt with using extra sensors

or estimation and filtering techniques according to the specific application's specifications and needs. In inertial navigation for example, where the gyroscope's noise causes a significant attitude error [22] over time, it is common to use a Kalman filter [11, 19] or use additional sensors, such as a magnetometer [10, 2], to get a more accurate attitude estimation.

A class of IMUs, known as gyroscope-free inertial measurements units (GF-IMUs), use several accelerometers mounted on a rigid object to provide an estimation of both the specific force and the angular velocity of the object. When the accelerometers are spread in a great distance GF-IMUs provide a very accurate, low-noise estimation of the angular velocity compared to a gyroscope of the same grade.

However, GF-IMUs come with a big disadvantage, which makes them inappropriate for real-world applications. More specifically, GF-IMUs cannot compensate for the accelerometers' bias; even a small bias on the accelerometers' measurement is translated into a constant drift in the estimated angular velocity. This is an important restriction for real-world applications as even if the sensor's bias is removed using a calibration procedure, a small drift of the bias is expected over time.

In this work we introduce an IMU architecture which combines several three-axis accelerometers and a single three-axis gyroscope, in a closed-loop configuration, to effectively reduce the measurement noise of the estimated angular velocity. The proposed architecture advances the concept of GF-IMUs and uses the measurements of a single three-axis gyroscope to dynamically compensate for the accelerometers' bias. The stability of the proposed system is examined analytically and closed-form conditions are provided. A model for the angular velocity noise is provided while simulations reveal the superior noise performance of the proposed system; in the upper frequency range, it presents more than 30dB less angular velocity noise compared to the gyroscope alone.

The rest of this work is structured as follows. In Section 8.2 the basic operation principles of the gyroscope-free inertial measurement systems are introduced while their main performance limitations are highlighted. In Section 8.3, the proposed closed-loop architecture is introduced and analyzed in detail. Stability conditions and a closed-form model for the system's noise are provided. In Section IV, the system's performance is tested and some important design considerations are expressed. Finally, conclusions are drawn in Section V.

8.2 Gyroscope-Free Inertial Measurement Units

In this Section, the basic principle of GF-IMUs is introduced and their performance limitations are highlighted.

8.2.1 Principle of Operation

Consider N single-axis accelerometers, placed arbitrary positions, $r_i, i = 1, 2, \dots, N$ on a rigid body and denote their sensitivity axes and measurements as $\hat{\eta}_i$ and f_i , respectively. Following [5], we write the following system of equations for deriving the specific force (f) and the angular

velocity (ω)

$$F = Jx + P \quad (8.1)$$

where

$$x = \begin{bmatrix} \dot{\omega} \\ f \end{bmatrix}, \quad J = \begin{bmatrix} J_1^T & J_2^T \end{bmatrix} \quad (8.2)$$

$$F = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}, \quad P = \begin{bmatrix} \hat{\eta}_1^T \Omega^2 r_1 \\ \hat{\eta}_2^T \Omega^2 r_2 \\ \vdots \\ \hat{\eta}_N^T \Omega^2 r_N \end{bmatrix}$$

the auxiliary variables J_1 and J_2 are

$$J_1 = \begin{bmatrix} (r_1 \times \hat{\eta}_1) & (r_2 \times \hat{\eta}_2) & \dots & (r_N \times \hat{\eta}_N) \end{bmatrix} \quad (8.3)$$

$$J_2 = \begin{bmatrix} \hat{\eta}_1 & \hat{\eta}_2 & \dots & \hat{\eta}_N \end{bmatrix}$$

and Ω is the cross-product matrix of the vector $\omega \triangleq [\omega_x \ \omega_y \ \omega_z]^T$

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (8.4)$$

Given an adequate number of properly placed (single-axis) accelerometers, one can solve (8.1) in a least squares sense and derive x as

$$x = (J^T J)^{-1} J^T (F - P) \quad (8.5)$$

Further defining $\bar{J} = (J^T J)^{-1} J^T$, (8.5) is written in a compact form as

$$x = \bar{J}F - \bar{J}P \quad (8.6)$$

and the solution is only valid if $J^T J$ is non-singular.

In this work, we focus on the solution of the system of differential equations for deriving the angular velocity, ω . Denoting the i^{th} row of \bar{J} as \bar{J}_i , we write

$$\dot{\omega} = \hat{J}F - \hat{J}P \quad (8.7)$$

where

$$\hat{J} = \begin{bmatrix} \bar{J}_1^T & \bar{J}_2^T & \bar{J}_3^T \end{bmatrix}^T \quad (8.8)$$

8.2.2 Existing Art and Performance Limitations

Several GF-IMU architectures have been proposed over the years. Many authors have proposed different configurations using six [5], nine [24, 9, 20], ten [16] or twelve [23, 8] (single-axis) accelerometers in an effort to provide a feasible solution to (8.7). Moreover, existing works use very specific geometries for the accelerometers' placement in an effort to simplify the original non-linear problem (8.7) for estimating the angular velocity. By doing so, the non-linear terms of (8.7) are eliminated and the derivative of the angular velocity is derived as a linear combination of the accelerometers' measurements as follows

$$\dot{\omega}_l = \hat{J}F \quad (8.9)$$

While existing works provide a very simple and computationally light solution to the original non-linear problem, the analysis is limited to the case of ideal accelerometers and neglects the effects of noise, bias and other imperfections of a real-world accelerometer. Since the bias is the largest contributor in the accelerometer's measurement error [21], we will examine the effect of a small additive bias, δF on the accelerometers' measurements. In this case, (8.9) becomes

$$\dot{\tilde{\omega}}_l = \hat{J}(F + \delta F) \quad (8.10)$$

where δF is the $N \times 1$ vectors representing the accelerometers' bias. Subtracting (8.9) from (8.10) we get the evolution of the system's output error in time

$$\delta \dot{\omega}_l \triangleq \dot{\omega}_l - \dot{\tilde{\omega}}_l = \hat{J}\delta F \quad (8.11)$$

As seen in (8.11), the output error of the existing systems accumulates over time implying that even a very small offset in the accelerometers' measurements causes a cumulative angular velocity error. This is rather important as even if the static sensors' offset is removed by a calibration procedure, a small offset drift is expected over time, even in the case of high-end sensors.

8.3 The Proposed System

In this Section, the proposed inertial measurement system is introduced and analyzed in detail. The stability of the proposed system under the effects of the accelerometers' and gyroscope's biases is investigated. Finally, an analytical expression of the output angular velocity noise is derived.

8.3.1 System Architecture

To alleviate the performance limitations of existing architectures, presented in Section 8.2.2, the proposed system uses multiple single-axis accelerometers in combination with a single three-

axis gyroscope in a closed loop configuration. The top-level architecture of the proposed system is shown in Figure 8.1.

In Figure 8.1, we consider the general case of an arbitrary configuration of $N > 6$ single-axis accelerometers and thus, the non-linear feedback term $\hat{J}P$ of (8.6) is also included. A second feedback-loop, compares the angular velocity calculated using the accelerometers' measurements and the GF-IMU theory with the measurement of the gyroscope and forces the lower-frequency part of the system's output (ω) to be equal to the gyroscope's measurement. In higher frequencies, the feedback signal attenuates and the system outputs the angular velocity estimated using the accelerometers' measurements. By doing so, the proposed system, ensures that the constant accelerometers' bias does not affect the system's output while in the upper frequency range the system outputs the low-noise angular velocity estimation derived using the accelerometers' measurements.

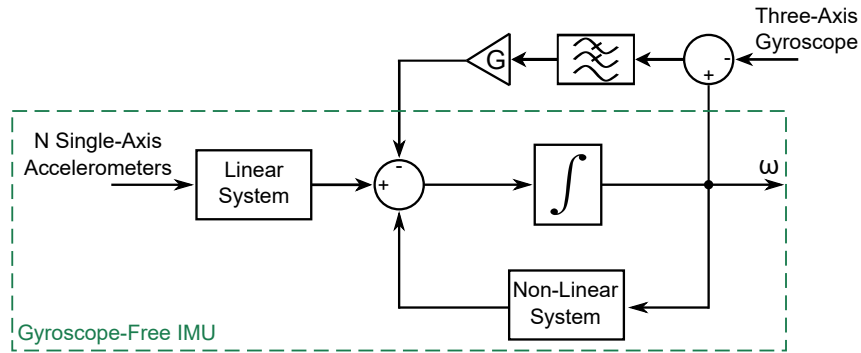


Figure 8.1: Top-level architecture of the proposed system.

8.3.2 The linear case

Before analyzing the general case, described by (8.7), in which the system is non-linear, it is helpful to consider the existing GF-IMU architectures. In this particular case, the proposed system is described by the block diagram of Figure 8.2 and the angular velocity is derived as the solution of a linear system of differential equations

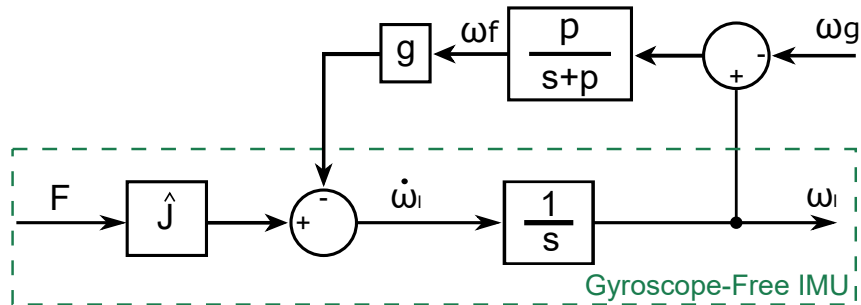


Figure 8.2: Block diagram representation of the proposed system for specific accelerometers' configurations eliminating the non-linear terms of (8.7).

At this point, since both the inputs (F and ω_g) and the output (ω_l) of the system are vectors,

it is useful to define the following diagonal matrices related to the modeling of the feedback loop.

$$\begin{aligned} G &= I_3 \otimes g \\ P_p &= I_3 \otimes p \\ P_s &= I_3 \otimes \left(\frac{p}{s+p} \right) \end{aligned} \quad (8.12)$$

where I_3 is the 3×3 identity matrix and \otimes denotes the Kronecker's product [25].

According to Figure 8.2, the system's output is

$$\dot{\omega}_l(F, \omega_g) \triangleq \hat{J}F - G \omega_f \quad (8.13)$$

Now, we assume non-ideal measurements for both the accelerometer and the gyroscope, and we define

$$F = F_i + \delta F \text{ and } \omega_g = \omega_{gi} + \delta \omega_g \quad (8.14)$$

where $F_i \in \mathbb{R}^6$ and $\omega_{gi} \in \mathbb{R}^3$ are the ideal measurements of the the accelerometers and the gyroscope respectively and $\delta F \in \mathbb{R}^6$ and $\delta \omega_g \in \mathbb{R}^3$ represent a small additive bias. Following the block diagram of Figure 8.2, the output error of the system is captured by the following system of differential equations

$$\begin{aligned} \dot{\delta \omega}_l &\triangleq \dot{\omega}_l(F_i + \delta F, \omega_{gi} + \delta \omega_g) - \dot{\omega}_l(F_i, \omega_{gi}) \\ &= \hat{J}\delta F - G\omega_f \end{aligned} \quad (8.15)$$

The feedback signal, ω_f , is written as

$$\begin{aligned} \omega_f &= P_s(\omega_l - \omega_g - \delta \omega_g) \\ &= P_s(\delta \omega_l - \delta \omega_g) \end{aligned} \quad (8.16)$$

and its time representation is derived as

$$\dot{\omega}_f = -P_p \omega_f + P_p(\delta \omega_l - \delta \omega_g) \quad (8.17)$$

Using (8.15) and (8.17), we write the following state-space system representation

$$\underbrace{\begin{bmatrix} \dot{\delta \omega}_l \\ \dot{\omega}_f \end{bmatrix}}_{\dot{x}_l} = \underbrace{\begin{bmatrix} 0_{3 \times 3} & -G \\ P_p & -P_p \end{bmatrix}}_{A_l} \underbrace{\begin{bmatrix} \delta \omega_l \\ \omega_f \end{bmatrix}}_{x_l} + \underbrace{\begin{bmatrix} \hat{J} & 0_{3 \times 3} \\ 0_{3 \times 6} & -P_p \end{bmatrix}}_{B_l} \underbrace{\begin{bmatrix} \delta F \\ \delta \omega_g \end{bmatrix}}_{u_l} \quad (8.18)$$

The characteristic polynomial of A_l is

$$p_{A_l}(\lambda) = (\lambda^2 + p\lambda + gp)^3 \quad (8.19)$$

and its roots (which are the eigenvalues of A_l) are negative for positive values of g and p . Thus A_l is Hurwitz and the system of (8.18) is BIBO stable. This is an important result as it indicates that the output error of the proposed system, $\delta\omega_l$, is bounded for bounded inputs (δF and $\delta\omega_g$).

To quantify the effect of the accelerometers' and the gyroscope's biases on the system's output error, we assume a small constant bias vector $\bar{f} \in \mathbb{R}^6$ for the accelerometers and a small constant bias vector $\bar{\omega} \in \mathbb{R}^3$ for the gyroscope. From (8.18) we get

$$\begin{aligned} x_l(t) &= e^{A_l t} x_l(0) + \int_0^t e^{A_l(t-s)} B_l \begin{bmatrix} \bar{f} \\ \bar{\omega} \end{bmatrix} ds \\ &= e^{A_l t} x_l(0) + (e^{A_l t} - I_6) A_l^{-1} B_l \begin{bmatrix} \bar{f} \\ \bar{\omega} \end{bmatrix} \end{aligned} \quad (8.20)$$

where I_6 is the 6×6 identity matrix. The steady state response of (8.18) is derived as

$$\begin{aligned} \lim_{t \rightarrow +\infty} x(t) &= -A^{-1} B \begin{bmatrix} \bar{f} \\ \bar{\omega} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\hat{J}\bar{f}}{g} + \bar{\omega} \\ \frac{\bar{f}}{g} \end{bmatrix} \end{aligned} \quad (8.21)$$

and consequently,

$$\delta\omega \Big|_{t \rightarrow +\infty} = \frac{\hat{J}\bar{f}}{g} + \bar{\omega} \quad (8.22)$$

The result of (8.22) is quite interesting as it indicates that in steady-state, the proposed system's offset is composed of a small portion of the accelerometers' bias and the whole gyroscope's bias. This comes in agreement with our intuition about the system's operation; the feedback loop forces the system's output to be equal to the gyroscope's one in low frequencies. Using the triangle inequality and (8.21), we get the worst case scenario for the steady state value of $\delta\omega$ which is

$$\|\delta\omega\| \leq \left\| \frac{\hat{J}\bar{f}}{g} \right\| + \|\bar{\omega}\| \quad (8.23)$$

and represents the case when the effect of the accelerometers' and the gyroscope's bias is additive.

8.3.3 The General, Non-Linear Case

To expand the results derived in Section 8.3.2 to the general case, when the proposed system is non-linear, it is useful to write (8.7) in a more convenient form. To do so, we begin by denoting the n^{th} row of matrix P as P_n , $n = 1, 2, \dots, N$

$$P_n = -\omega_x^2 \alpha_n - \omega_y^2 \beta_n - \omega_z^2 \gamma_n + \omega_x \omega_y \delta_n + \omega_x \omega_z \epsilon_n + \omega_y \omega_z \zeta_n \quad (8.24)$$

where

$$\begin{aligned}
\alpha_n &= \eta_n^y r_n^y + \eta_n^z r_n^z \\
\beta_n &= \eta_n^x r_n^x + \eta_n^z r_n^z \\
\gamma_n &= \eta_n^x r_n^x + \eta_n^y r_n^y \\
\delta_n &= \eta_n^x r_n^y + \eta_n^y r_n^x \\
\epsilon_n &= \eta_n^x r_n^z + \eta_n^z r_n^x \\
\zeta_n &= \eta_n^y r_n^z + \eta_n^z r_n^y.
\end{aligned} \tag{8.25}$$

The product $\hat{J}P$ in (8.7) is written as

$$\hat{J}P = \begin{bmatrix} - & - & \bar{J}_1 & - & - \\ - & - & \bar{J}_2 & - & - \\ - & - & \bar{J}_3 & - & - \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_N \end{bmatrix} \tag{8.26}$$

which, by replacing P_1, P_2, \dots, P_n from (8.24) and after some algebraic manipulation becomes

$$\hat{J}P = \underbrace{\begin{bmatrix} \bar{J}_1 A & \bar{J}_1 B & \bar{J}_1 \Gamma & \bar{J}_1 \Delta & \bar{J}_1 E & \bar{J}_1 Z \\ \bar{J}_2 A & \bar{J}_2 B & \bar{J}_2 \Gamma & \bar{J}_2 \Delta & \bar{J}_2 E & \bar{J}_2 Z \\ \bar{J}_3 A & \bar{J}_3 B & \bar{J}_3 \Gamma & \bar{J}_3 \Delta & \bar{J}_3 E & \bar{J}_3 Z \end{bmatrix}}_M \underbrace{\begin{bmatrix} -\omega_x^2 \\ -\omega_y^2 \\ -\omega_z^2 \\ +\omega_x \omega_y \\ +\omega_x \omega_z \\ +\omega_y \omega_z \end{bmatrix}}_{L(\omega)} \tag{8.27}$$

where

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix}, B = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_N \end{bmatrix}, \Gamma = \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_N \end{bmatrix} \tag{8.28}$$

$$\Delta = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_N \end{bmatrix}, E = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}, Z = \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_N \end{bmatrix}$$

Now, considering the proposed system of Figure 8.1 and replacing (8.27) into (8.7), we write

$$\dot{\omega} = \hat{J}F - ML(\omega) - G\omega_f \tag{8.29}$$

where the filter's dynamics, similarly to (8.17), are described by

$$\dot{\omega}_f = -P_p \omega_f + P_p(\delta\omega - \delta\omega_g) \quad (8.30)$$

and the ground truth angular velocity is given by

$$\dot{\omega}_{gi} = \hat{J}F - ML(\omega_{gi}) \quad (8.31)$$

We formulate the output error's dynamics as in the linear case. Considering $\delta\omega = \omega - \omega_{gi}$, we have

$$\dot{\delta\omega} = -M(L(\omega) - L(\omega_{gi})) + \hat{J}\delta F - G\omega_f \quad (8.32)$$

By substituting $\omega = \delta\omega + \omega_{gi}$ we get

$$\begin{aligned} L(\omega) - L(\omega_{gi}) &= K(\omega_{gi})\delta\omega + L(\delta\omega) \\ &= K(\omega_{gi})\delta\omega + O(\|\delta\omega\|^2) \end{aligned} \quad (8.33)$$

where

$$K(\omega_{gi}) = \begin{bmatrix} -2\omega_{gi}^x & 0 & 0 \\ 0 & -2\omega_{gi}^y & 0 \\ 0 & 0 & 2\omega_{gi}^x \\ \omega_{gi}^y & \omega_{gi}^x & 0 \\ \omega_{gi}^z & 0 & \omega_{gi}^x \\ 0 & \omega_{gi}^z & \omega_{gi}^y \end{bmatrix}. \quad (8.34)$$

Neglecting the higher order terms in (8.33), we rewrite (8.32) as

$$\dot{\delta\omega} = -MK(\omega_{gi})\delta\omega + \hat{J}\delta F - G\omega_f \quad (8.35)$$

Using (8.35) and (8.17), we write the following state-space system representation for the proposed system

$$\underbrace{\begin{bmatrix} \dot{\delta\omega} \\ \dot{\omega}_f \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} -MK(\omega_{gi}) & -G \\ P_p & -P_p \end{bmatrix}}_A \underbrace{\begin{bmatrix} \delta\omega \\ \omega_f \end{bmatrix}}_x + \underbrace{\begin{bmatrix} \hat{J} & 0_{3 \times 3} \\ 0_{3 \times 6} & -P_p \end{bmatrix}}_B \underbrace{\begin{bmatrix} \delta F \\ \delta\omega_g \end{bmatrix}}_u \quad (8.36)$$

Comparing (8.36) with (8.18) we notice that the only difference is the the North-West block in matrix A . This block depends on the actual angular velocity vector ω_{gi} and is of course time-varying. As a result, we have to further investigate the stability of the autonomous part of the system (i.e. $\dot{x} = Ax$) which is strongly related to BIBO stability [13, 15]. To this end, and since the system of (8.36) can be seen as quasi-linear, we touch upon on some well-established results in Linear Parameter Varying (LPV) system's theory [4, 27, 18].

To start with, we define the time-varying parameters $\delta_i \triangleq \omega_{gi}^i$ for $i = 1, 2, 3$ and so as A is a

parameter varying matrix (i.e. $A \triangleq A(\delta_1, \delta_2, \delta_3)$). These parameters appear in an affine way in A , i.e.

$$A = A_0 + A_1\delta_1 + A_2\delta_2 + A_3\delta_3 \quad (8.37)$$

Considering that the angular velocity of the object is bounded, the parameters δ_i , $i = 1, 2, 3$ are also considered to be bounded, i.e. $|\delta_i| \leq \delta_{max}$. Hence, the parameter vector $\delta = [\delta_1 \ \delta_2 \ \delta_3]^\top \in \mathcal{D} = \text{co}(\delta^1, \delta^2, \dots, \delta^8)$, with $\delta^j \in \mathbb{R}^3$ for $j = 1, 2, \dots, 8$ where $\text{co}(\cdot)$ denotes the convex hull of vertices δ^j [3]. In our case, \mathcal{D} composes a cube centered at zero with vertices $\{\pm\delta_{max}, \pm\delta_{max}, \pm\delta_{max}\}$. Furthermore, we define the set of vertices $\mathcal{D}_l = \{\delta^1, \delta^2, \dots, \delta^8\}$.

The parametric varying system $\dot{x} = A(\delta)x$ with $\delta \in \mathcal{D}$ is exponentially stable if [4] $\exists \ X \in \mathbb{R}^{6 \times 6}, X \succ 0$ ¹:

$$A^\top(\delta)X + XA(\delta) < 0 \quad \forall \delta \in \mathcal{D} \quad (8.38)$$

Using (8.38), one can in theory prove the stability of the proposed system by solving an infinite number of Linear Matrix Inequalities (LMI). In our case however, the stability conditions can be relaxed and use only a finite amount of LMIs [4, 18]. Considering that $A_l(\delta)$ is affine on parameter vector δ , the parametric varying system $\dot{x} = A(\delta)x$ with $\delta \in \mathcal{D}$ is exponentially stable if $\exists \ X \in \mathbb{R}^{6 \times 6}, X \succ 0$:

$$A^\top(\delta)X + XA(\delta) < 0 \quad \forall \delta \in \mathcal{D}_l. \quad (8.39)$$

The origin is an exponentially stable equilibrium point for the nonlinear system (8.32) if it is an exponentially stable equilibrium point for the linear system $\dot{x} = Ax$ [13]. Thus exponential stability for the nonlinear dynamic is also ensured in a local sense.

The stability condition derived in (8.39) implies an arbitrary time-varying parameter δ . If the rate of variation of δ is bounded, the aforementioned stability test is conservative. In our system, δ corresponds to the angular velocity of an object, and its rate of change can in most cases be considered to be bounded.

Similarly, we define $\dot{\delta} \triangleq [\dot{\delta}_1 \ \dot{\delta}_2 \ \dot{\delta}_3]^\top \in \mathcal{D}' = \text{co}(\dot{\delta}^1, \dot{\delta}^2, \dots, \dot{\delta}^8)$ where $\dot{\delta}^j$ denotes the vertices of a cube centered at zero (i.e. $\dot{\delta}^j \in \{\pm\dot{\delta}_{max}, \pm\dot{\delta}_{max}, \pm\dot{\delta}_{max}\}$). We consider $\dot{\delta}$ to be bounded, i.e. $|\dot{\delta}_i| \leq \dot{\delta}_{max}$ and we define the set of vertices $\mathcal{D}'_l = \{\dot{\delta}^1, \dot{\delta}^2, \dots, \dot{\delta}^8\}$. The autonomous part of (8.36), $\dot{x} = Ax$, is exponentially stable in the large, if there exist X_0, X_1, \dots, X_p with $A_\nu^\top X_\nu + X_\nu A_\nu \succeq 0$, $\nu = 1, 2, \dots, m$ [18] such that

$$\sum_{k=0}^m X_k \delta_k \succ 0 \quad (8.40)$$

and

$$\sum_{k=1}^m X_k \dot{\delta}_k + \sum_{\nu=0}^m \sum_{\mu=0}^m \delta_\nu \dot{\delta}_\mu (A_\nu^\top X_\mu + X_\mu A_\nu) \prec 0 \quad (8.41)$$

$\forall \delta \in \mathcal{D}_l$ and $\forall \dot{\delta} \in \mathcal{D}'_l$ and $\delta_0 = 1$. In this case, $m = 3$ (8.37).

The set of LMIs described in (8.39) and (8.40) - (8.41) can be solved using any Semi Definite

¹ $\succ 0$ denote a positive-definite matrix.

Programming Suite (e.g. SeDuMi [26]) in standard computational platforms such as Python (e.g. cvxpy [6]), Matlab (e.g. Yalmip [17]) etc.

To sum up, in the linear case (8.18) which corresponds to particular configurations of the accelerometers, the stability of the proposed system is easily ensured (8.19) and only small attention is required on the design of the feedback's filter. However, for an arbitrary configurations of the accelerometers we notice that there is a nonlinear feedback term (8.6) which introduces a time-varying block as shown in (8.36). This implies that the stability of the system relies on the sensors' configuration. On the one hand, there is more freedom for the sensor structure to be selected but on the other hand one has to be careful to guarantee stability for the custom configuration.

8.3.4 Output Noise Modeling

To examine the output angular velocity noise, we consider a linearization of the proposed system around zero. The linearized system is described by the block-diagram of Figure 8.3

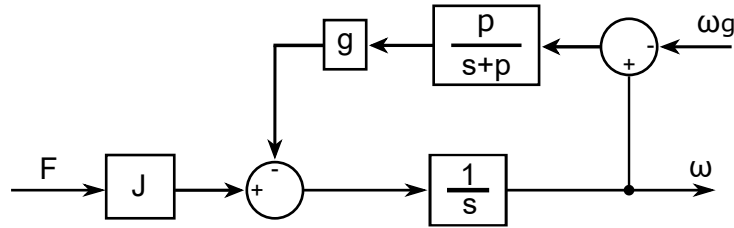


Figure 8.3: Block diagram representation of the proposed system linearized around $\omega = 0$ rad/s.

Since the system is linear, we use superposition to calculate the output angular velocity as a function of the two inputs: the accelerometers' measurements (F) and the gyroscope's measurement (ω_g). More specifically, it is:

$$\omega(s) = (I_3 \otimes H_F(s)) \hat{F}(s) + (I_3 \otimes H_g(s)) \omega_g(s) \quad (8.42)$$

where

$$\begin{aligned} H_F(s) &= \frac{s+p}{s^2+ps+gp} \\ H_g(s) &= \frac{gp}{s^2+ps+gp} \end{aligned} \quad (8.43)$$

and

$$\hat{F}(s) = JF(s) \quad (8.44)$$

Denote the power spectral density (PSD) of the gyroscope's measurement as $S_g(s)$ and the PSD of \hat{F} as $S_{\hat{F}}(s)$. Assuming that the output noise of the accelerometers and the gyroscope are uncorrelated, the PSD of the output angular velocity, S_ω , is derived as follows

$$S_\omega = (I_3 \otimes |H_F(s)|^2) S_{\hat{F}}(s) + (I_3 \otimes |H_g(s)|^2) S_g(s) \quad (8.45)$$

where

$$\begin{aligned} |H_F(s)|^2 &= \frac{-s^2 + p^2}{s^4 + (2gp - p^2)s^2 + g^2p^2} \\ |H_g(s)|^2 &= \frac{g^2p^2}{s^4 + (2gp - p^2)s^2 + g^2p^2} \end{aligned} \quad (8.46)$$

The exact characteristics of the accelerometers' and the gyroscope's noise depend on the sensor used and are different even along sensors using the same manufacturing technology. However, typically, the noise of the accelerometers and the gyroscope is considered to be white noise. This is a reasonable assumption used in many works to facilitate the mathematical analysis and in most cases gives accurate results.

We assume that both the inputs F and ω_g are excited with white noise, i.e.

$$F \sim \mathcal{N}(0, C_F) \text{ and } \omega_g \sim \mathcal{N}(0, C_\omega) \quad (8.47)$$

where C_F and C_ω denote the covariance of the accelerometers' and gyroscope's noise respectively. The noise of the linear combination of the accelerometers' measurements, JF , is also white noise with covariance $C_{JF} = JC_FJ^T$ [1] and thus it is

$$\hat{F} \sim \mathcal{N}(0, C_{JF}) \quad (8.48)$$

In this case, the PSD of the output angular velocity, S_ω , is derived as follows

$$S_\omega = (I_3 \otimes |H_F(s)|^2) (J \circ J) S_F + (I_3 \otimes |H_g(s)|^2) S_g \quad (8.49)$$

where S_F is the PSD of the accelerometers' noise and \circ denotes the Hadamard's product [12].

8.4 Results

In this section we use a simple configuration, shown in Figure 8.4, composed of nine single-axis accelerometers (grouped in three three-axis ones) and a single three-axis gyroscope to evaluate the performance of the proposed system. In addition, we highlight the restrictions imposed by the previously derived stability conditions to the sensors' placement and the design of the feedback filter.

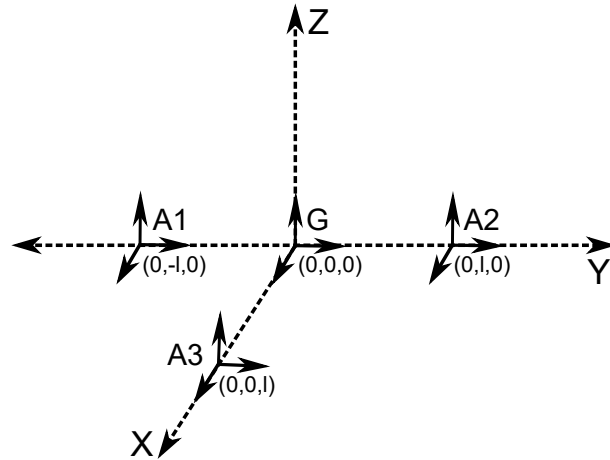


Figure 8.4: IMU configuration composed of nine single-axis accelerometers (grouped in three three-axis ones) and a single three-axis gyroscope.

8.4.1 Noise Model Accuracy

Before we evaluate the performance of the proposed system, we test the accuracy of the derived noise model of Section 8.3.4. To do so, we consider the system configuration of Figure 8.4 and assume the sensors' distance to be equal to $l = 1m$. The feedback's gain is set to $g = 10$ while the cut-off frequency of the low-pass filter is set to $p = 6\pi$ rads (3 Hz).

Using MATLAB's Simulink we simulated the described configuration assuming white noise sequences for both the accelerometers' and the gyroscope's inputs. The power spectral density (PSD) of the output noise for all accelerometers is set to be equal to $S_A = -50dB/Hz$ while the PSD of all gyroscope's axes is $S_G = -65dB/Hz$. In Figure 8.5, the PSD of the system's output resulted from the simulation is compared to the one calculated using the noise model (8.49).

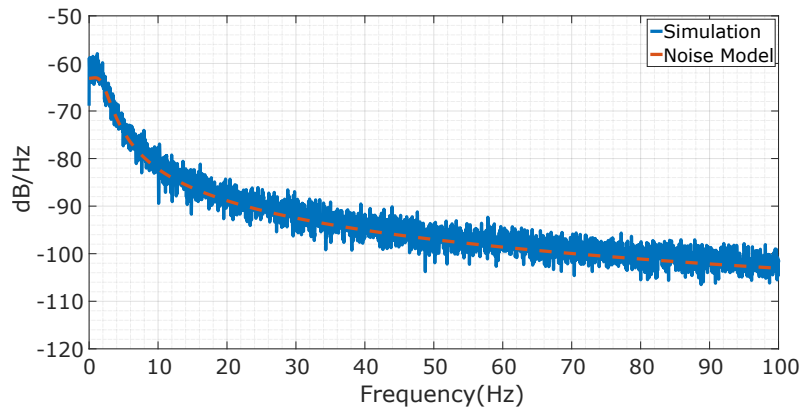


Figure 8.5: The PSD of the output of the proposed system (x-axis) compared to the PSD calculated using the noise model of (8.49).

8.4.2 Noise Performance

To evaluate the noise performance of the proposed system, we used the sensors' configuration of Figure 8.4 and excited both the accelerometers and the gyroscope with white noise. The noise characteristics and the feedback loop design parameters were set similar to Section 8.4.1, i.e. $S_G = -65\text{dB}/\text{Hz}$, $S_A = -50\text{dB}/\text{Hz}$, $g = 10$ and $p = 6\pi$ rads.

The PSD of the system's output is presented in Figure 8.6 and compared to the PSD of the gyroscope's noise for two different values of the parameter l in Figure 8.4. As seen in Figure 8.6, while the distance between the accelerometers gets greater, the output noise of the proposed system gets significantly lower in the higher frequencies where the output is dominated by the accelerometers' measurements.

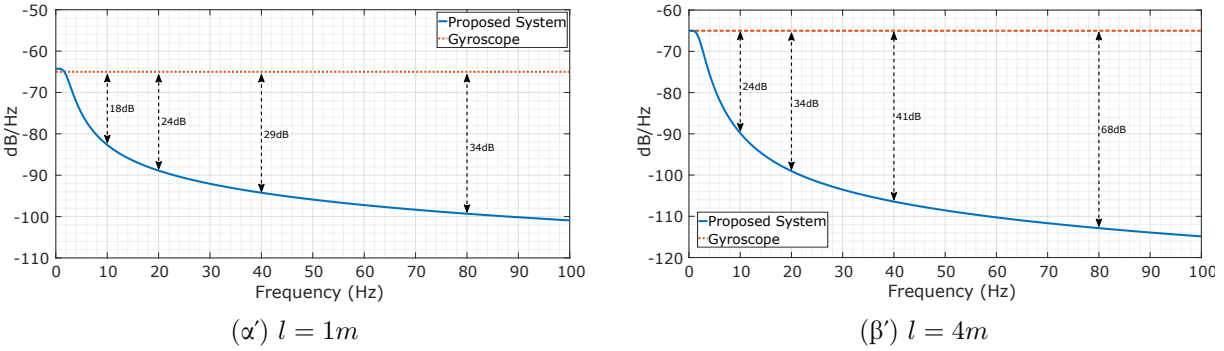


Figure 8.6: The PSD of the proposed system's output noise (X-axis) compared to the PSD of the gyroscope's output noise (X-axis) for $l = 1m$ (a) and $l = 4m$ (b).

8.4.3 System Design and Stability Considerations

In Section 8.4.2 we demonstrated that the angular velocity noise of the proposed IMU becomes lower when the accelerometers are spread over a wider distance. In this Section we will examine the effect of the feedback filter's design on the output noise and furthermore we will demonstrate how improper design of the filter may lead to an unstable system.

We assume the configuration of Figure 8.4 and we consider the sensors' distance to be equal to $l = 2m$. The power of the system's angular velocity noise in a 100Hz bandwidth is presented in Tables 8.1-8.4 for different values of the maximum angular velocity (ω_{max}) and the feedback filter's pole ($f_p = p/2\pi$) and gain (g) parameters. The unstable configurations according to (8.39) are denoted with "X" in Tables 8.1-8.4.

In Table 8.1, for $\omega_{max} = 1\text{rad}/\text{s}$, the system is stable for every pair (f_p, g) and the filter's design only affects the output noise power which varies from -61.9dB up to -53.3dB . Note the lowest noise power of -61.9dB is more than 15dB lower than the output noise of the gyroscope in the same bandwidth. While the value of ω_{max} gets higher, the stability of the system depends more on the design of the feedback's filter. However, even in the case of $\omega_{max} = 10\text{rad}/\text{s}$ (Table 8.4), proper design of the feedback filter leads to a stable configuration with only 1.4dB more noise power (-60.5dB) which is still about 15dB lower than the output noise of the gyroscope

in the same bandwidth.

	g = 5	g = 10	g = 20	g = 50
f_p = 0.2Hz	-58.0dB	-57.4dB	-56.1dB	-53.3dB
f_p = 1Hz	-60.9dB	-59.8dB	-57.6dB	-54.1dB
f_p = 2Hz	-61.4dB	-60.2dB	-57.9dB	-54.2dB
f_p = 5Hz	-61.9dB	-60.5dB	-58.0dB	-54.3dB
Gyroscope	-45.3dB			

Table 8.1: Angular velocity noise within $[0, 100]Hz$ for $\omega_{max} = 1rad/s$ and different values of f_p and g .

	g = 5	g = 10	g = 20	g = 50
f_p = 0.2Hz	X	X	X	X
f_p = 1Hz	-60.9dB	-59.8dB	-57.6dB	-54.1dB
f_p = 2Hz	-61.4dB	-60.2dB	-57.9dB	-54.2dB
f_p = 5Hz	-61.9dB	-60.5dB	-58.0dB	-54.3dB
Gyroscope	-45.3dB			

Table 8.2: Angular velocity noise within $[0, 100]Hz$ for $\omega_{max} = 5rad/s$ and different values of f_p and g .

	g = 5	g = 10	g = 20	g = 50
f_p = 0.2Hz	X	X	X	X
f_p = 1Hz	X	X	X	-54.1dB
f_p = 2Hz	X	-60.2dB	-57.9dB	-54.2dB
f_p = 5Hz	X	-60.5dB	-58.0dB	-54.3dB
Gyroscope	-45.3dB			

Table 8.3: Angular velocity noise within $[0, 100]Hz$ for $\omega_{max} = 7rad/s$ and different values of f_p and g .

	g = 5	g = 10	g = 20	g = 50
$f_p = 0.2\text{Hz}$	X	X	X	X
$f_p = 1\text{Hz}$	X	X	X	X
$f_p = 2\text{Hz}$	X	X	-57.9dB	-54.2dB
$f_p = 5\text{Hz}$	X	-60.5dB	-58.0dB	-54.3dB
Gyroscope	-45.3dB			

Table 8.4: Angular velocity noise within $[0, 100]Hz$ for $\omega_{max} = 10rad/s$ and different values of f_p and g .

References

- [1] Theodore W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley-Interscience, 2003.
- [2] J. Bird and D. Arden. “Indoor navigation with foot-mounted strapdown inertial navigation and magnetic sensors [Emerging Opportunities for Localization and Tracking]”. In: *IEEE Wireless Communications* 18.2 (2011), pp. 28–35. DOI: 10.1109/MWC.2011.5751293.
- [3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2013.
- [4] Corentin Briat. *Linear Parameter-Varying and Time-Delay Systems. Analysis, Observation, Filtering and Control*. Jan. 2015. ISBN: 978-3-662-44049-0.
- [5] Jeng-Heng Chen, Sou-Chen Lee, and Daniel B. DeBra. “Gyroscope free strapdown inertial measurement unit by six linear accelerometers”. In: *Journal of Guidance, Control, and Dynamics* 17.2 (1994), pp. 286–290. DOI: 10.2514/3.21195. URL: <https://doi.org/10.2514/3.21195>.
- [6] Steven Diamond and Stephen Boyd. “CVXPY: A Python-embedded modeling language for convex optimization”. In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5.
- [7] E. Dorveaux et al. “Iterative calibration method for inertial and magnetic sensors”. In: *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. Dec. 2009, pp. 8296–8303. DOI: 10.1109/CDC.2009.5399503.
- [8] Ezzaldeen Edwan, Stefan Knedlik, and Otmar Loffeld. “Constrained Angular Motion Estimation in a Gyro-Free IMU”. In: *IEEE Transactions on Aerospace and Electronic Systems* 47.1 (2011), pp. 596–610. DOI: 10.1109/TAES.2011.5705694.
- [9] Joseph Genin, Juehui Hong, and Wei Xu. “Accelerometer Placement for Angular Velocity Determination”. In: *Journal of Dynamic Systems, Measurement, and Control* 119.3 (Sept. 1997), pp. 474–477. ISSN: 0022-0434. DOI: 10.1115/1.2801281. URL: <https://doi.org/10.1115/1.2801281>.
- [10] Paul Groves et al. “Inertial Navigation Versus Pedestrian Dead Reckoning: Optimizing the Integration”. In: vol. 2. Sept. 2007.
- [11] Paul D. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2013.
- [12] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2013.
- [13] Hassan Khalil. *Nonlinear Systems*. Pearson, 2001.
- [14] M. Kok et al. “Calibration of a magnetometer in combination with inertial sensors”. In: *2012 15th International Conference on Information Fusion*. July 2012, pp. 787–793.

- [15] Douglas Leith and W.E. Leithead. "Survey of gain-scheduling analysis and design". In: *Int. J. Control* 73 (Jan. 2000), pp. 1001–1025. DOI: 10.1080/002071700411304.
- [16] An Li et al. "Gyroscope Free Strapdown Inertial Navigation System Using rotation modulation". In: *2009 Second International Conference on Intelligent Computation Technology and Automation*. Vol. 3. 2009, pp. 611–614. DOI: 10.1109/ICICTA.2009.613.
- [17] J. Löfberg. "YALMIP : A Toolbox for Modeling and Optimization in MATLAB". In: *In Proceedings of the CACSD Conference*. Taipei, Taiwan, 2004.
- [18] Javad Mohammadpour Velni and Carsten Scherer. *Control of Linear Parameter Varying Systems with Applications*. Jan. 2012. ISBN: 9781461418337. DOI: 10.1007/978-1-4614-1833-7.
- [19] Aboelmagd Noureldin, Tashfeen B. Karamat, and Jacques Georgy. *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*. Springer-Verlag Berlin Heidelberg, 2013.
- [20] A. J. Padgaonkar, K. W. Krieger, and A. I. King. "Measurement of Angular Acceleration of a Rigid Body Using Linear Accelerometers". In: *Journal of Applied Mechanics* 42.3 (Sept. 1975), pp. 552–556. DOI: 10.1115/1.3423640. URL: <https://doi.org/10.1115/1.3423640>.
- [21] K. Papafotis and P. P. Sotiriadis. "MAG.I.C.AL.—A Unified Methodology for Magnetic and Inertial Sensors Calibration and Alignment". In: *IEEE Sensors Journal* 19.18 (Sept. 2019), pp. 8241–8251. ISSN: 1530-437X. DOI: 10.1109/JSEN.2019.2919179.
- [22] Konstantinos Papafotis and Paul P. Sotiriadis. "Exploring the Importance of Sensors' Calibration in Inertial Navigation Systems". In: *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2020, pp. 1–4. DOI: 10.1109/ISCAS45731.2020.9181212.
- [23] Fang-Jun Qin, Jiang-Ning Xu, and An Li. "A Novel Attitude Algorithm for 12 Accelerometer Based GFINS Using Hermite Interpolation". In: *2010 International Conference on Measuring Technology and Mechatronics Automation*. Vol. 1. 2010, pp. 214–217. DOI: 10.1109/ICMTMA.2010.414.
- [24] Fangjun Qin, Jiangning Xu, and Jiang Sai. "A New Scheme of Gyroscope Free Inertial navigation System Using 9 Accelerometers". In: *2009 International Workshop on Intelligent Systems and Applications*. 2009, pp. 1–4. DOI: 10.1109/IWISA.2009.5072652.
- [25] Gilbert Strang. *Linear Algebra and Its Applications*. Brooks Cole/Cengage Learning, 2007.
- [26] Jos F Sturm. "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones". In: *Optimization methods and software* 11.1-4 (1999), pp. 625–653.
- [27] Roland Toth. "Modeling and Identification of Linear Parameter-Varying Systems". In: vol. 403. Jan. 2010. DOI: 10.1007/978-3-642-13812-6.

-
- [28] J. F. Vasconcelos et al. “Geometric Approach to Strapdown Magnetometer Calibration in Sensor Frame”. In: *IEEE Transactions on Aerospace and Electronic Systems* 47.2 (Apr. 2011), pp. 1293–1306. ISSN: 0018-9251. DOI: 10.1109/TAES.2011.5751259.
- [29] Y. Wu and W. Shi. “On Calibration of Three-Axis Magnetometer”. In: *IEEE Sensors Journal* 15.11 (Nov. 2015), pp. 6424–6431. ISSN: 1530-437X. DOI: 10.1109/JSEN.2015.2459767.

9

Conclusions and Further Research

In the first part of this Thesis, a complete methodology for calibrating and aligning the sensitivity axes of a 3-axis magnetometer, a 3-axis accelerometer and a 3-axis gyroscope was introduced. The proposed methodology compensates for the most significant linear, time-invariant error sources of inertial and magnetic field sensors and requires no special piece of equipment to apply. Both simulation and experimental results prove that the proposed methodology performs very well in terms of convergence, repeatability and computational efficiency.

Some interesting research directions stemming from this work are listed below:

- This work only considers the most important linear, time invariant error sources for the three sensors. The proposed algorithm could be expanded to also compensate for non-linear error sources (e.g. scale factor non-linearity) or dynamic, time-varying phenomena such as the temperature dependence of the sensors' measurements.
- The calibration of inertial and magnetic field sensors without using special calibration equipment is a very popular research topic. Many works present different algorithms that successfully calibrate inertial and magnetic field sensors using a set of measurements in different sensors' orientations. However, the measurement acquisition methodology is typically derived experimentally by the authors and different works propose different methodologies without the related mathematical proof that the acquired measurements are sufficient to solve the (minimization or estimation) problem of calibration. Thus, a very interesting research topic is the measurement acquisition procedure for inertial and magnetic field sensors calibration and the related mathematical analysis.

In the second part of the Thesis, two applications of inertial and magnetic field sensors are presented. First, a pedestrian navigation algorithm is introduced. Using shoe-mounted inertial and magnetic field sensors, the proposed algorithm accurately estimates the velocity, position and orientation of a walking person. It exploits the characteristics of the human walking and uses the TRIAD algorithm along with the popular zero-velocity update method to ensure the long-term accuracy of the orientation estimation. Experimental results demonstrated the accuracy and the long-term stability of the proposed algorithm.

Based on this work, the following research directions could be explored:

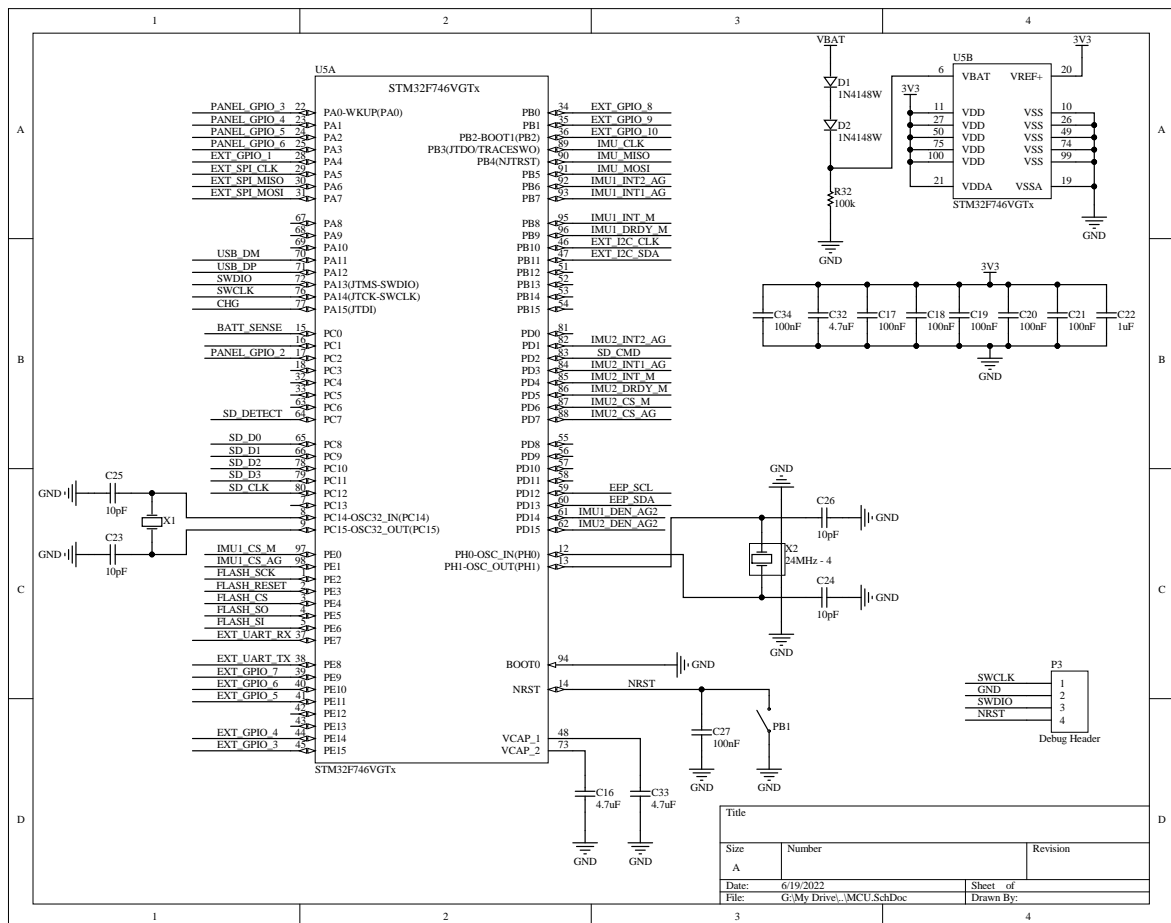
- The proposed algorithm excels in terms of computational efficiency compared to similar algorithms using complicated heading reference systems. However, using only inertial and magnetic field sensors, its deployment is limited to outdoor environments with limited sources of magnetic disturbance. An expansion of the proposed algorithm that would also exploit information from other sources such as RF beacons, building maps, etc. would enable its wider deployment in indoor environments.

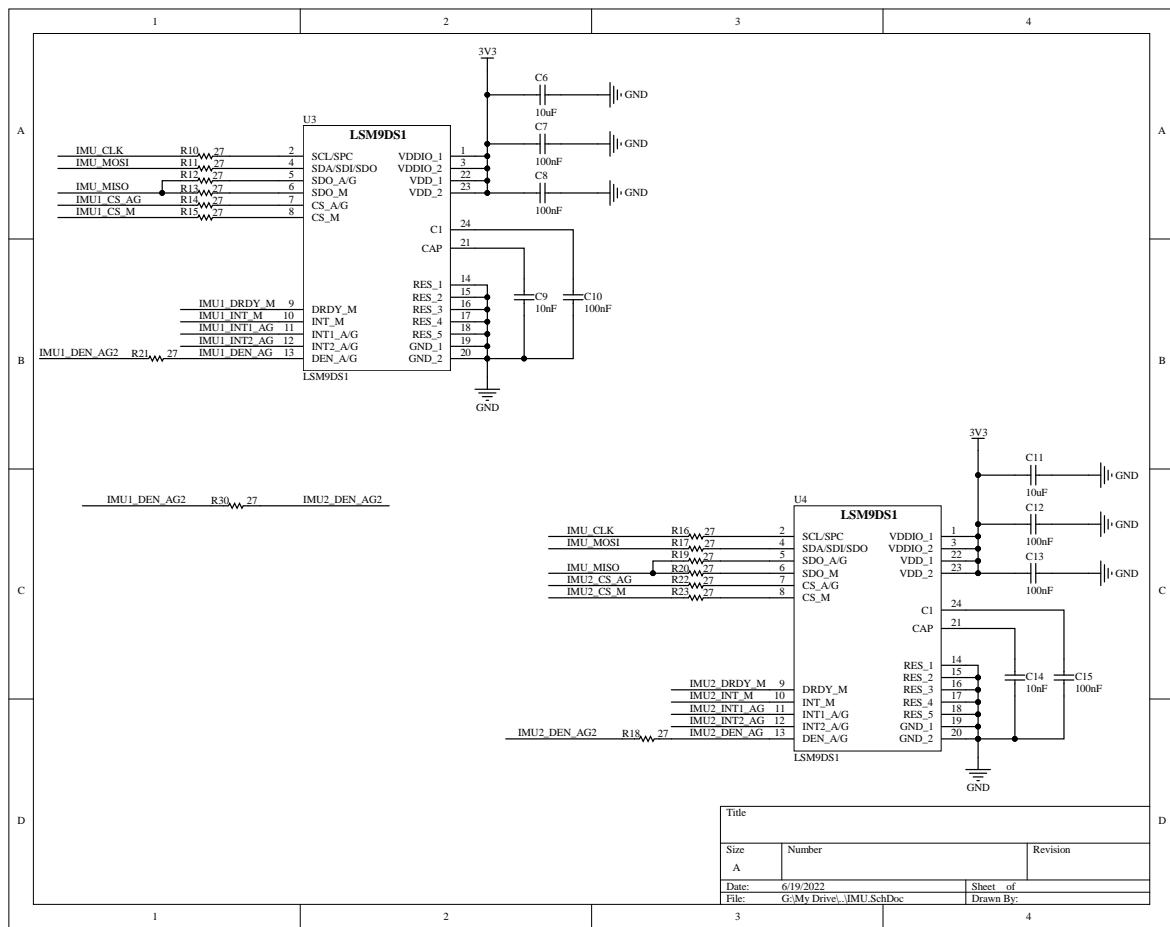
An IMU architecture using several accelerometers and a single three-axis gyroscope was also introduced in the second part of the Thesis. The proposed architecture combines the low-noise characteristic of GF-IMUs and ensures the systems stability and immunity to the accelerometers' bias by using the gyroscope in a closed-loop configuration. Extensive theoretical analysis as well as simulation results indicated that the proposed system is capable of providing up to $15dB$ less angular velocity noise in its output compared to a gyroscope of the same grade while its stability can be guaranteed when it is carefully designed. GF-IMUs were extensively studied over the past years but their inherent disadvantage to compensate for the accelerometers' bias made them inappropriate for real-world applications despite their very good noise performance. The proposed architecture provides a solution to this problem and enables the development of low-noise, high-performance inertial measurement units.

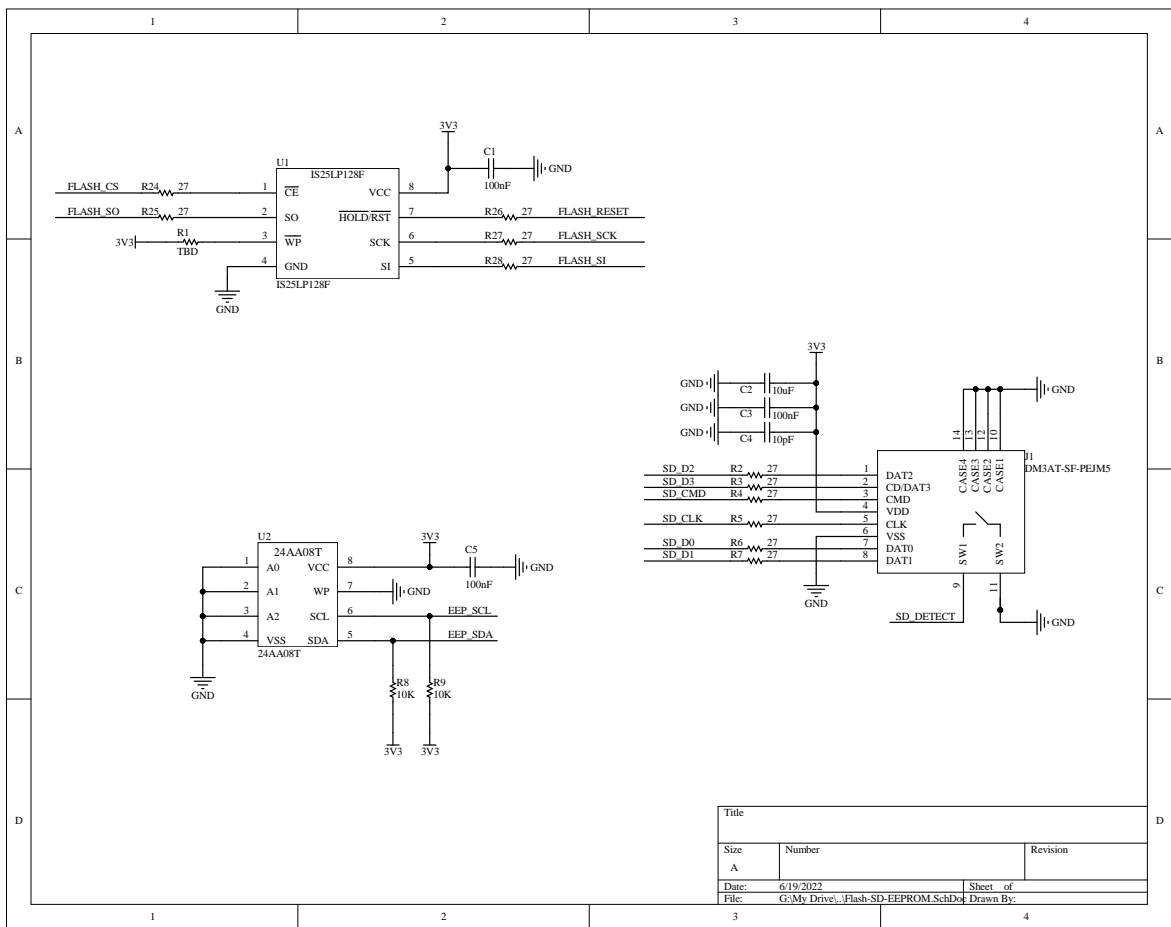
- In this work, the resilience of the proposed system to the accelerometers' and gyroscope's bias is investigated. The effect of other, common, sensors' errors such as cross-axis sensitivity and noise could also be explored. In addition, instead of requiring the sensors' to be pre-calibrated, algorithms that could calibrate the sensors after the system's deployment could be developed.
- The proposed algorithm uses the sensors' distance and orientation from a reference coordinate frame to calculate the object's angular velocity. An interesting topic of research would be the effect of the sensors' placement error on the angular velocity estimation. As a solution to this problem, smart algorithms that detect the sensors' placement geometry and eliminate the need for accurate sensor's placement could be developed. This would guarantee the accuracy of the angular velocity estimation while it would also make easier the real-world deployment of the system.

10

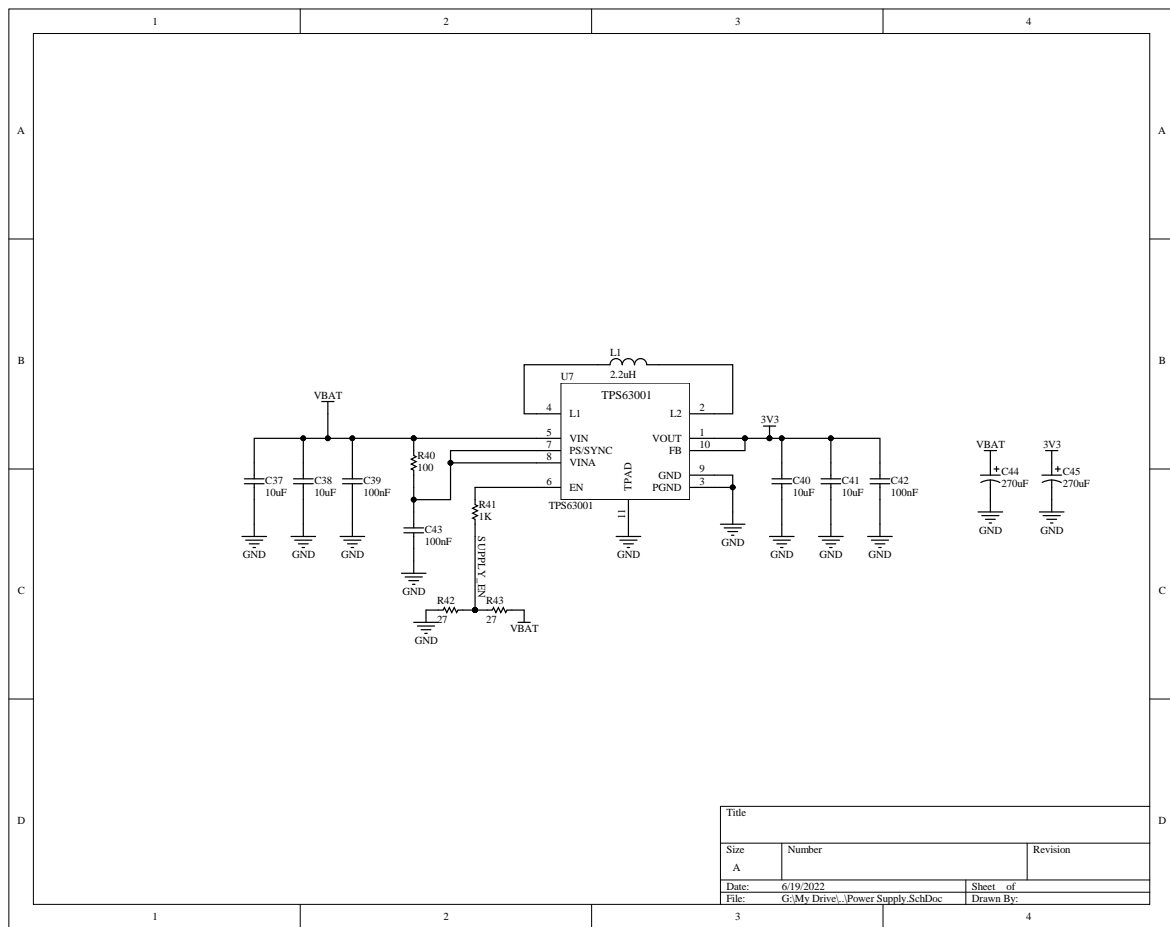
Appendix A: IMU Schematics







Title		
Size	Number	Revision
A		
Date:	6/19/2022	Sheet of
File:	G:\My Drive\Flash-SD-EEPROM SchDoc Drawn By:	



Title		
Size	Number	Revision
A		
Date:	6/19/2022	Sheet of
File:	G:\My Drive\...Power Supply_SchDoc	Drawn By:

