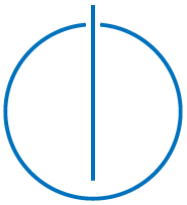
Technische Universität München

Hybrid sensor fusion with dual quaternion-based EKF for pose estimation

Charalampos Papathanasis

Master's Thesis
in
Biomedical Computing
Department of Informatics



Technische Universität München
Department of Informatics

Master's Thesis in Biomedical Computing

Hybrid sensor fusion with dual quaternion-based EKF for pose estimation

Eine multimodale Sensorfusionsmethode mit Dual-Quaternion EKF zur Posenbestimmung

Charalampos Papathanasis

Advisors:

Prof. Dr. Nassir Navab
Federico Tombari, Ph.D
Benjamin Busam, M.Sc.

Submission date: 15 June 2018

Declaration of Authorship

I, Charalampos Papathanasis, confirm that this Master's thesis is my own work and I have documented all sources and material used.

This thesis was not previously presented to another examination board and has not been published.

Signed: _____

Date: _____

"A new scientific truth does not triumph by convincing its opponents and making them see the light, but rather because its opponents eventually die, and a new generation grows up that is familiar with it."

Max Karl Ernst Ludwig Planck (1858 - 1947)

Abstract

The field of computer vision and more specifically the pose estimation pipelines developed heavily over the last decade [1],[2],[3],[4]. Modern sensors of various technologies are used in several solutions from everyday life to medical applications and space engineering. In this project, our goal is to design an algorithm that estimates the poses of the object(s) being tracked through the input of more than one optical or inertial sensors. We use an efficient and compact dual quaternion formalization to capture the coupling between the translation and rotation and to construct an Extended Kalman Filter (EKF) in order to fuse information from different modalities e.g. IMU and OTS. With this method, we are able to overcome the disadvantages of each sensor so as to obtain more accurate results and bridge occlusion intervals. The filter is implemented for up to four sensors and can be extended to a generic number of sensors. The model is evaluated using synthetic and real data for different occasions including the occlusion case where the line-of-sight is lost for the vision sensor. We achieve a reduction of the position error for the synthetic data from 2.14 m of the vision sensor alone to 1.68 m with the addition of an IMU sensor and further reduction to 1.56 m with the fusion of four sensors. Analogously the angle error for the rotation was decreased from 20.13° of the vision sensor to 10.15° for the fusion of two sensors and 9.22° for the four-sensor fusion. Regarding the occlusion case with the real data, the filter for an occlusion of 3 seconds obtains a position error of 2.21 m at the end of the occlusion, compared to the mean error of 1.25 m with standard deviation of 0.31 m for the whole trajectory. The according values for the angle error are 3.96° at the end of the occlusion in comparison with the mean angle error of 1.53° with standard deviation of 0.97° for the whole trajectory.

Acknowledgements

I take this opportunity to express gratitude to my supervisors for their help and support, especially to Mr Busam. Without his passionate participation and input, the survey could not have been successfully conducted. I also want to thank FRAMOS GmbH, Germany for giving me the opportunity to accomplish the thesis in their R&D department and providing me with the necessary equipment. Finally I deeply thank my parents, friends and partner for the unceasing encouragement and support.

Contents

List of Figures

List of Tables

To my family and partner

Chapter 1

Introduction

1.1 Motivation

Nowadays there are several tracking technologies that have been developed and there is a great scientific and technological interest about the applications that involve these kind of devices. Several sensors have been implemented in the past years, all with its advantages and disadvantages. As the needs for more accurate and robust tracking systems are increasing because of the great variety of applications, sensor fusion is becoming a more common way to overcome the difficulties arising from the drawbacks of individual sensor technologies.

Especially in the biotechnological and medical disciplines, there are various systems that are based on tracking technologies and sensors, and there is an increased interest for further development of more efficient and better solutions[10]. Optical tracking systems (OTS) are of the most widespread kind of sensor technologies because of their accuracy and they are used in many medical applications[11],[12][13]. Also, inertial measurement units (IMU) are well known and common sensors since they are easy to use and they provide reliable information about the orientation of the object they are attached to. Furthermore the use of Electromagnetic Tracking Systems which are relatively new, are of clinical importance as they are used to support the already existing tracking systems[14].

The motivation for this project is the plurality of the cases where tracking sensors are used in hybrid or individual systems and the necessity for more efficient

ways to use this kind of technologies not only for computer aidded medical procedures but also for other occasions such as augmented reality setups, mobile device applications, robotics, autonomous flying/driving etc.

1.2 Aims and objectives

As mentioned above the individual sensors do have certain specifications, they have their own logic and a specific way of operation. For example, optical tracking systems use markers that are tracked by an optical tracker and therefore they need the so called line of sight in order to capture the exact position of the object(s). This is obviously the major drawback of an OTS apart maybe from its high cost. On the other hand, the IMU sensors are sufficient to provide information about the orientation but when it comes to the position estimation they suffer from major drift.

The objective goal of this project is to find a way to estimate both the orientation and the position of the target object(s) by the fusion of all provided information from multiple sensors. This fusion is accoplished by implementing a filter which compensates the drawbacks of the different sensors and as it can handle the information of multiple modalities the output is more accurate. The line of sight issue of the OTS is reduced by the information that is provided by the other sensors and of course the position errors from the IMU is corrected by the measurements of the OTS and the modeled process as we describe in detail in the next chapters.

As we will show in section 1.4 there are several research groups that have dealt with this problem and the research for this topic is still ongoing.

1.3 Tracking technologies

In this section we briefly introduce the main characteristics and hardware principles along with the advantages and disadvantages of the most common tracking devices.

1.3.1 Optical Tracking Systems

Optical tracking systems proved to be an early answer to clinically feasible tracking systems and for many other applications. They evolved into the most reliable and accurate tracking solution. OTS usually consist of charge-coupled device (CCD) cameras and sequentially illuminated infrared (IR) light-emitting diodes (LED). The IR light is reflected by retro-reflective markers, that can be detected in space and then can be reconstructed by reconstruction algorithms in the tracking system [15].These are called passive markers. There are also active markers which they emit IR photons which are detected by the camera sensor. There are two approaches in OTS technology:

- Inside-out tracking: the camera is mounted on the tracked device or object and the markers are placed in stationary locations.
- Outside-in tracking: the camera(s) is/are placed in a stationary location and the markers are attached to the tracked object.

It is also feasible to perform marker-less tracking by identification, classification and comparison of the tracked objects from pre-defined patterns or 3D models[16].

There are many advantages that the OTS have and the most important one is that they provide accurate data and are less susceptible to noise. Also they do not suffer from drift problems, they are wireless and can track multiple objects inside the measurement space. This is why OTS is well established and used in many applications.

The major limitation of this kind of tracking system is the line of sight issues. If somebody or something interfere between the camera and the marker then we loose information and the object can no longer be tracked. This is a big drawback for example in computer-aided medical applications. Other than that, the calibration of the OTS can be challenging and the cost of the system is relatively higher than other tracking devices.

In the following figure 1.1 a representation of the sensor camera and passive markers is shown:

FIGURE 1.1: Polaris camera with passive marker tool[5]

1.3.2 Inertial Measurement Units

The inertial measurement unit (IMU) uses data from accelerometers, gyroscopes and sometimes also magnetometers that are integrated in a single device. The acceleration measure the linear acceleration, the gyroscope the angular velocity and the magnetometer the direction of the magnetic field[17].With the development of Micro-Electro-Mechanical systems [18] the IMUs became cheaper, more efficient and yet more popular in various applications such as GPS systems, autonomous flying/driving and airspace[19],[20]. With the data from the IMU we can indirectly calculate the both the orientation via the gyro measurements and the position by integrating twice the acceleration. This is the major drawback of the use of the IMU for position estimation as the error of the measurements is cumulative and through time this leads to drift. In general the IMUs are not used solely as tracking devices but more in combination with other tracking devices.

On the other hand the IMUs have many advantages. Firstly they are not affected by external influences except for the changes in gravity. As mentioned above they are cheap and they can be integrated easily in a tracking system in order to give good estimations about orientation. Furthermore IMUs are lightweighted and small so they can easily be mounted in the tracked object(s).

In figure 1.2 a typical IMU is shown.

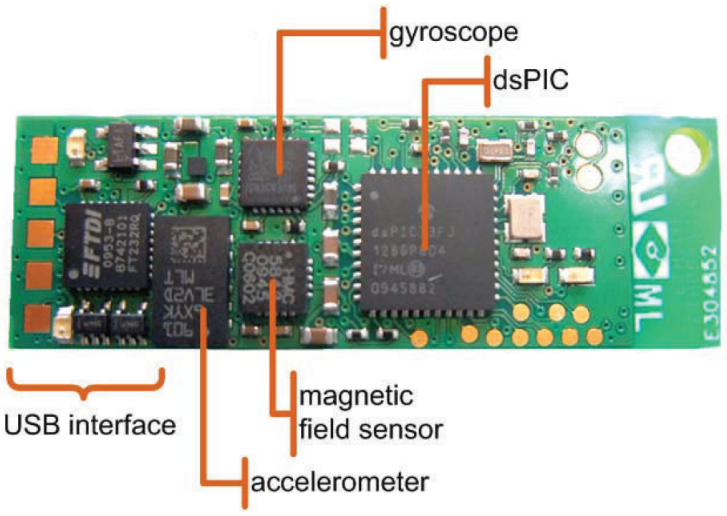


FIGURE 1.2: Inertial Measurement Unit with all its components[6]

1.3.3 Electromagnetic tracking systems

Electromagnetic tracking systems (EMTS) are a relatively new tracking technology. Their main advantage is that they have no line-of-sight limitation, but their disadvantages include susceptibility to distortion from nearby metal sources and limited accuracy compared to optical tracking. These systems localize small electromagnetic field sensors in an electromagnetic field of known geometry. The EMTS can be divided into three categories as described below[15]:

- AC-driven tracking. The earliest developed classical EMTS are driven by alternating current (AC). This system consists of three coils arranged in a Cartesian coordinate system that emits an electromagnetic field composed of three dipole fields. Typical operating frequencies for the AC-driven magnetic trackers lie in the range of 814 kHz. Small search coils measure the induced voltage, which is proportional to the flux of the magnetic field.

- DC-driven tracking. As the name would suggest, rather than using an AC-driven magnetic field, these systems are driven by quasistatic direct current (DC). The magnetic induction within miniature active (fluxgate) sensors was originally measured after establishment of a stationary magnetic field, but current models employ passive microminiaturized sensors.

- Passive or transponder systems. These systems track position by localization of permanent magnets or implanted transponders.

In the figure 1.3 the Aurora AC-based Electromagnetic tracking system is shown (courtesy of Northern Digital)

FIGURE 1.3: Aurora AC-based Electromagnetic tracking system[7]

1.4 Related work

The idea of combining sensors data is for surely not a new one, it is a broad research subject in the past years and the studies about it are numerous [21],[22], [23],[24],[25]. In this section we will focus on the literature and the research that is done for the combination of inertial and optical sensor data sources can be optical tracking systems (OTS) or cameras that are used for estimation of the pose of the tracked object(s), via sophisticated algorithms for this purpose (e.g. SLAM or EM) [9],[26]. The papers that are referenced use different approaches, mathematical techniques and filters for the fusion of the data from the different sensors and propose solutions in various fields and applications. Based on these references we try to develop a method that will overcome the known drawbacks and introduce the multiple sensor fusion so to evaluate its use in potential applications.

1.4.1 Inertial data & marker-based vision system

There are some publications about the fusion of inertial and marker-based vision system data (OTS, cameras with fiducial markers). The motivation for all these studies is of course to provide accurate results in a robust way and to resolve issues such as line of sight for the vision system or errors due to drift for the estimation of position from the IMU measurements. Furthermore, the goal is a higher sampling rate with the aid of the inertial sensor and also the avoidance of singularities with the use of quaternions for the representation of the orientation.

Enayati et al.(2015) [27] implement an Unscented Kalman Filter (UKF) to estimate the position and orientation of a freely moving object for surgical applications. With the use of an OTS for computer-assisted surgery (bandwidth of 20-60 Hz) and measurements from a 200 Hz IMU they performed two experiments, one for frequency augmentation of the two sensors and the other one for the case of marker occlusion of the OTS. They use the UKF which was introduced by J. Julier, Simon and K. Uhlmann [28], instead of the more popular EKF and they claim that the first does not have the known disadvantages of the EKF approach[29], even though there are studies that suggest that the EKF is computationally more efficient with slightly worse performance than the UKF (Mathieu et al.(2004) [29], Armesto et al.(2007) [24]). The concept is that for every 10 estimated samples from IMU, one sample from the OTS is acquired to correct the drift. By this fusion technique they achieve a 10 times faster hybrid system with accuracy of below 1mm for position and 1° for orientation that satisfies the requirements of tracking in surgical applications[30].

An experiment with an augmented reality head-mounted display that integrates an OTS and an IMU is performed by He et al.(2015) [2]. The experiment is divided in two cases, one under partial occlusion of the markers and one under total occlusion. The output of the optical tracking unit is used to estimate the bias of the IMU sensors.The cosine algorithm is used to calculate the true value of gravity and magnetic field, which is orthogonal to the value of gravity from the Euler angles obtained from the camera data. The bias measurement is estimated by subtracting these true values from the sensor feedback, and then using a Finite Impulse Response (FIR) filter to attenuate the noise. If the marker points are occluded and there are no orientation results from the OTS, the bias estimation process is stopped and parameters of the bias model are not updated. The most recent bias model is used until the markers become visible again. Under partial occlusion conditions they have accurate results for both orientation and position estimation. When it comes to the total occlusion they have only small errors when the occlusion lasts for less than a second. If the occlusion last longer, then the drift is more severe and the algorithm diverges from the ground truth. This filter may give good results under long term partial occlusion and short term total occlusion.

Hartmann et al.(2010) [31] made an indoor 3D position tracking with an IMU and a marker based video tracking system with external cameras. Position, velocity and attitude are calculated from the IMU measurements and fused by using the

position corrections from the visual tracking system. The system was tested with both synthetic and real data and proved to give accurate results, dealing with the drift problems from the IMU and the measurement flaws from the video. An extended Kalman Filter was implemented for the fusion process and typical Euler angles were used as mathematical formalization for orientation despite the well-known drawbacks[32]. Without having a solid ground truth, the biggest differences between EKF estimations and video tracking system measurements were in the range of a few centimeters. The filter was able to overcome short time occlusions and the lack of the video tracking system when the tracked object was moving rapidly.

Another study for the pose estimation using sensor data fusion with optical tracker and IMU for image guided surgery applications is done by Oh et al.(2015) [4]. The hardware setup for the proposed system involves one IMU fixed to the OTS and another IMU fixed to the surgical tool with a marker attached. This is arranged in order to calibrate the sensors. During the process of calibration the authors eliminate the accelerometer DC bias of IMU by subtracting the stable state values from the output values of the accelerometer along the three axis of coordinate.After that they align the coordinate axis by using the fact that gravity applied to both IMUs is equal. Then they implement a linear Kalman Filter to fuse the data. When occlusion occurs, the estimation is done by only the IMU measurement correction process, independent from the OTS measurement correction process. In this particular study they show results of the orientation of the instrument as they claim that this is the most important component for this kind of application. With the specific calibration method, they are able to have a maximum divergence rate of 0.0193°/s during occlusion.

Also, Tobergte et al.(2009) [33] present an algorithm for pose estimation by sensor fusion. The method is based on an extended Kalman filter technique and is intended to fuse inertial and OTS data. In case of no or partial occlusion the filter is giving good results by increasing the accuracy of the system.The RMSE for position in X-axis is 0.12 mm and 0.0006 rad for the yaw in the fusion system rather than the according 0.83mm and 0.002 rad for the optical tracker. In the case of total occlusion, like other studies above, the position estimation drifts away after a couple of seconds.The RMSE error is becoming 29.2 mm for X-axis position, although the corresponding values for the yaw error is only 0.0009 rad. The advantage of the filter is that it couples the redundant sensors tightly using

information about every visible marker ball of the OTS for the estimation, while taking their different sampling rates and latencies into account.

Another paper that proposes a fusion method for AR registration is from You et al.(2001) [25]. In their implementation they use robust vision landmark tracker, inertial gyro sensors, and the complementary fusion filter. The fiducial system consists of calibrated landmarks that they use for the tracking and they make the landmark training offline. They quantitatively evaluate the system under dynamic conditions, and the experimental results show that the fusion method achieves high tracking stability. For image projection the max tracking error is 9.8 pixels, the average error is 1.84 pixels and the error covariance is 5.32 pixels.

1.4.2 Inertial and vision for marker-less applications

Except for the use of systems using cameras and markers for tracking, researchers also use marker-less methods to obtain poses so as to combine them with inertial data. In this paragraph we will briefly introduce some studies that use mono, stereo or more cameras and IMU data for various applications.

Kumar et al.(2014) [1] are trying to solve the tracking problem for mobile augmented reality (MAR) applications by using a fusion system. They implement an EKF to combine the data from the IMU sensor, so as to calculate the orientation and then another EKF to fuse these data with the pose acquired from vision through an Adaptive Meanshift algorithm. The parameters of the filter are obtained by extensive experimentation and they performed two different experiments, one with occlusion and one without it. Their figures show that the fusion results appear to be smooth and closely following the vision measurements than the IMU and the system can overcome the occlusion problem through short periods of time (about half a second).

A study about applications in minimally invasive surgery with sensor fusion was done by Giannarou et al.(2012)[34]. The goal was a robust framework for intraoperative free-form deformation recovery based on structure from-motion. For that they implemented a UKF to fuse vision information with IMU data. Validation of the system was done with synthetic and phantom data, with the ground truth

poses taken by an OTS. They claim to have much better results for the deformation recovery with the use of the fusion algorithm both in the synthetic and phantom data and so a potential clinical value.

Armesto et al.(2007)[24] is dealing with ego-motion estimation by fusion of vision and inertial data, using both EKF and UKF and doing a performance comparison of the two filters. They present an approach of multi-rate EKF and multi-rate UKF to deal with data from different sampling rates. The results show that both filters give nearly the same results for selected covariance values and therefore their responses overlap. Also, they state that for this particular application, EKF gives better performance than UKF, since both filters provide nearly the same estimation, but the computational cost of UKF is about 7 times higher. Furthermore, the fusion of inertial and vision gives better performance results than single estimations. Fusion introduces more benefits to pure inertial than to pure vision estimation but this is mainly due to the double integration performed on the inertial measurements where bias correction can not be performed.

A recent application of tracking a racket trajectory in real time was presented by Zhang et al.(2017)[3]. The main idea of this approach lies in tracking the trajectory of the racket on a players hand by fusion of vision and IMU sensors data. For the vision data they use both monocular and binocular vision systems. With the use of a typical EKF they manage to combine the data from the already fused data of the IMU (from accelerometer, gyroscope and magnetometer) and the racket pose from the image processing of the binocular and monocular vision systems. The racket position from fused data was more accurate than when it was calculated only via visual measurement. As for the racket orientation the fusion method neutralized the data of visual measurement and the IMU. The average angle error is for yaw 0.8981°, for roll 1.6331° and for pitch 0.7818°.

A master thesis from Hugmark(2013)[26] aimes for the development of a system that enables the overlay of computer graphics in a video sequence recorded by a moving camera, to be used for augmented reality in a dynamic environment. And for that he uses the monocular SLAM approach and an IMU to complement the visual input from the camera. He implementes an EKF to fuse the sensors input, and also the Madgwick algorithm[35] to get a more precise orientation estimation from the inertial measurements. He evaluates the system using multiple video sequences and the results indicates that using the combination of a camera and IMU sensor allows for more robust camera pose estimations during more difficult

circumstances. Examples of such circumstances include occlusions, feature-poor environments and diverse types of combined movements. The system has also some failures such as that the translations are sometimes not registered, especially in feature-poor environments. Another is that the system is very sensitive to synchronization-issues between the IMU and the camera.

Another study from Parnian et al.(2008)[21] deals with a position tracking system for a hand-held tool based on low cost sensors. The hardware that they use for this project was a multi camera vison system (four low cost CCD cameras in a curved line) and an MEMS-based inertial sensor. For the fusion of the data a modified EKF, also known as Indirect Kalman Filter, was implemented with which the error states of the system are applied to the system. Because of the multiple cameras they had to deal with a complicated calibration procedure to estimate the intrinsic and extrinsic camera parameters. A video tracking algorithm was used to estimate the 3D position of the tool tip by averaging 3D positions which are measured by each two adjacent cameras. The integration of vision system and inertial sensors for their experiments showed that the tracking of the object could be achieved with a higher sampling rate and less error compared to the typical stereo camera vision systems.

Hol et al.(2007)[22] use an EKF and UKF for fusion purposes. The main goal was the estimation and prediction of a camera position and orientation by fusion of the data from the inertial and vision sensors for AR applications. They found that the UKF has similar accuracy with the EKF but significantly higher computational cost, in accordance with Armesto et al.(2007)[24]. Their framework for non-linear estimation in real time, is able to track the camera with an absolute accuracy of 2cm for position and 1° for orientation. They concluded also that the addition of the IMU yields a more robust system which can handle periods of occlusion and reduce the need for high frequency vision updates.

Micro Aerial Vehicle (MAV) state estimation also is a field that uses sensor fusion. Weiss et al.(2013)[19] in their study try to develop a system that robustly works for a long duration (about 350 sec) in large, unknown and GPS-denied environments. So, they implemented a framework which uses feeds from monocular camera and an IMU to achieve real time and onboard autonomous flying. The popular EKF was used so as to combine effectively the measurements of the two sensors and they claim to have achieved autonomous flights of more than 360 m trajectory and 70 m altitude change followed by autonomous landing.

A study for motion tracking in unknown environments using visual and inertial measurements was done by Li and Mourikis(2013) [36]. They term this task as visual-inertial odometry (VIO), and basically, they have implemented a special case of EKF, a multi-state-constraint Kalman filter (MSCKF), so as to combine the data from an IMU and a monocular camera. Monte Carlo simulations and real world experimental testing show that the MSCKF algorithm attains better accuracy and consistency than EKF-based SLAM. The RMSE for position is 14.401 m and for orientaiton is 1,102° which are lower than the corresponding ones from 3 EKF-SLAM algorithms that were compare to. The system is capable of performing long-term, high-precision, consistent VIO in real time.

A similar study about VIO from Leutenegger et al.(2015)[37] introduces a framework of tightly-coupled fusion of inertial measurements and image keypoints in a nonlinear optimization problem that applies linearization and marginalization in order to achieve keyframing. Without the use of Kalman filtering they obtaine poses, velocities, and IMU biases as a time series, as well as a 3D map of sparse landmarks. They claim that their framework achieves high accuracy, while still being able to operate at real-time, despite the high computation cost.

1.4.3 IMU and vision with dual quaternions

The studies mentioned above use mostly the mathematical formalization of quaternions to express the rotation component so as to avoid singularities potentially occurred from the use of traditional Euler angles, and is well established. For the case of sensor fusion there is also some research with the mathematical approach that we are using in the current study, the dual quaternions. In the next sections these relevant papers are being briefly presented.

The most relevant study and the closest to our method is the one from Varghese et al.(2015)[38]. They are proposing a dual quaternion based EKF for IMU and vision data fusion for mobile AR applications. They suggest that the coupled translational and rotational motions are best calculated through the dual quaternion representation in comparison with the existing quaternion based algorithms. To verify their results, they make experiments with both synthetic and real data and their results show that the dual quaternions manage successfully to estimate the poses. However, there seems not to be much of a difference in terms of accuracy in rotation component and also, they do not take the case of occlusion of the

vision sensor into account. Furthermore because of the nature of the application they fuse the data only from the IMU sensor and the camera of the mobile. The method is compact and useful for this kind of applications as the position error covariance for the dual quaternion approach is smaller than the single quaternion method.

Another research study that is involved in the use of dual quaternions is the one from Schwaab et al(2016)[39]. In their work a simplified strapdown inertial mechanization is used to predict the orientation and position of the user while the measurements of visual odometry are employed to reduce the fast position drift of a pure inertial approach. The difference is that the authors are using the framework of stochastic cloning for both the rotation and translation measurements which are incorporated in a statistically consistent way including a proper treatment of the uncertainties of the previous estimates. For this purpose, an error-state extended Kalman filter is implemented and they claim that their method is useful for indoor and outdoor scenarios. The work also demonstrates how the fusion algorithm can be conveniently expressed in terms of dual quaternions, and their results from known datasets show an improvement in accuracy compared to the single sensor approaches which is however, not always significant. Also, they mention that the drift problems from the IMU measurements for the estimation of position cannot be eliminated completely rather than being minimized.

Yuyang et al.(2016)[20] propose an approach to position, velocity and attitude estimation for Micro Aerial Vehicles(MAVs) using dual quaternions. More precisely they use an EKF along with error propagation equations based on an additive error model. The results from the simulated data show that the dual quaternion can capture coupling between translation and rotation more accurately compared to the quaternion based method. Furthermore, it improves the precision of estimated results especially in the velocity and position estimation in altitude estimation. However, they do not examine any occlusion case, or data from datasets or real data.

Extensive research is done by Filipe et al.(2015)[40] for Spacecraft Pose Estimation Using Dual Quaternions. It is based in the previous Quaternion Multiplicative Extended Kalman Filter (Q-MEKF)[41] and they extend the method to Dual Quaternion Multiplicative Extended Kalman Filter (DQ-MEKF) for spacecraft pose (i.e., attitude and position) and linear and angular velocity estimation. They use the concept of error unit dual quaternion analogously to the concept of error

unit quaternion to develop a multiplicative EKF. The difference with most of the studies is that they use continuous-time angular velocity and linear acceleration measurements with noise and bias and discrete-time pose measurements with noise. The advantage of this approach is that the discrete-time measurements do not need to be equally spaced in time, making irregular or intermittent measurements easy to handle. According to the experimental results, the DQ-MEKF does not encounter singularities and is accurate, precise, and fast enough for operational use. Moreover, when compared with two other EKF formulations, experimental results and Monte-Carlo simulations suggest that the DQ-MEKF might be the best formulation if the measurements are expressed in a different reference frame than the variable to be estimated.

1.4.4 Other related work

In the literature there are also some research studies that verify some of the methods that we use in our approach, (i.e. use of dual quaternions formalization, alman filter for nonlinear systems) are worth to be mentioned and also some related work to different fusion techniques.

An early research study to evaluate the use of dual quaternions in pose estimation was done from J. S. Goddard and M. A. Abidi(1998)[42]. The general problem is to locate an object and measure its relative motion in three dimensions given a sequence of 2D intensity images of the object whose position and orientation are known relative to a base reference frame. The 3D transformation was modeled as a non-linear stochastic system with the state estimate providing the six degrees of freedom for the motion and position values as well as structure. The stochastic model uses the iterated EKF as the nonlinear estimator and the representation of the 3D transformation was based on dual quaternions. They test their method with both simulated and real experimental data. The results shown that the dual quaternion based IEFK estimates the poses more accurately than the previous used point based EKF.

Another scientific paper from Srivatsan et al.(2016)[43] look at the problem of estimating a time invariant SE(3) element for various applications such as registration, object tracking and sensor calibration. In their work, they use dual quaternions to represent the SE(3) element and use multiple measurements simultaneously to rewrite the measurement model in a truly linear form with state dependent

measurement noise. They show that their linear measurement model allows for decoupled estimation of rotation and translation using independent Kalman filters and the results suggest that the dual quaternion-based linear filtering is capable of estimating the SE(3) more accurately with less computation time compared to state-of-the-art filtering methods for SE(3) estimation (EKF,UKF).

Tao et al.(2005)[23] propose a method for home-based rehabilitation, in which the different data modalities from two sensors (vision and IMU) are fused by using arm structure relationship and geometry information without filtering for the estimation. Tracking performance of the proposed hybrid tracking system for arm motion is evaluated by comparing the results with ground truth data and they have shown that the system is able to track the arm movement in real time and accurately with very low computational cost. Of course the specific system is limited to this kind of applications.

1.5 Approach

Taking into consideration the previous work and research that is done and the different methodologies and mathematical approaches that have been used so far, we implemente an Extended Kalman Filter (EKF) based on a dual quaternion mathematical formalization, that can take as input more than two different modalities so that we have more information about the pose of the object(s).

The EKF is a well-established and well-known method when it comes to the fusion of different modalities and sensors, and has been around for over 20 years [1],[2], [38],[24],[25],[22],[33]. The use of the dual quaternion formalization has several advantages as it combines the rotation and translation of the object and this coupling gives a more precise estimation [38],[40],[20]. It may not be as intuitive as the quaternion for rotation and the standard translation vector but it is more compact and can be easily integrated in a system that is already quaternion-based. Furthermore, it has all the benefits of the use of the quaternions instead of using the classical Euler angles approach that has the major drawback of Gimbal lock. or the more computationally expensive rotation matrices. The calculations are faster and so the filter can be used for real time applications

1.6 Outline overview

In Chapter 2, we present all the theoretical background and research needed for this thesis. All the mathematical formalization and the algebra of the dual numbers, the quaternions and dual quaternions are introduced. The basic concept behind the implementation of the filter is also introduced.

In Chapter 3, we go into more detail about the algorithm and the implementation itself. The modeling of the process is shown and the prototype is extensively presented.

In Chapter 4, we show step by step how the algorithm is developed, the software and the tools that are used, and the flow of information.

In Chapter 5, the results and the evaluation by synthetic and real data, are being demonstrated along with the according graphs and figures.

In Chapter 6, the final chapter, we provide a summary of our work, we discuss the results and reach to a conclusion for the current project including open questions future work.

Chapter 2

Mathematical background

An introduction to the mathematical concepts of quaternions, dual numbers and dual quaternions is presented in this chapter. Also, we introduce the basic idea behind the Kalman filter and the Extended Kalman filter. These mathematical preliminaries are essential which is why the filter model is explicitly described in the next chapters.

2.1 Quaternions

The quaternions are a number system that represent an extension to the complex numbers. They were firstly introduced by William Rowan Hamilton in 1843 [44]. Back then, they were just a theoretical mathematical concept but in recent years they are broadly used in calculations of three-dimensional rotations due to their advantages over classical Euler angle representation [45]. The quaternions were the first non-commutative division algebra to be discovered [46],[47].

2.1.1 Definition

A single quaternion is expressed as:

$$q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$$

where a, b, c, d are real numbers and $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are orthogonal imaginary unit vectors.

Analogously to the imaginary numbers, $b\mathbf{i}+c\mathbf{j}+d\mathbf{k}$ is called the imaginary part or the vector part and a is the real part or scalar part. A quaternion that has only the imaginary part is called purely imaginary. It can also be represented as $q = [s, \mathbf{v}]$ where s is the scalar part and $\mathbf{v}$ the vector part, and also as $q = [\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\mathbf{v}]$ where θ is the angle of rotation and $\mathbf{v}$ is the axis of rotation.

The unit quaternions are defined by the following basic property:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$$

The equations below shows that the multiplication of quaternions is non-commutative as already mentioned above.

$$\begin{aligned} \mathbf{ij} &= \mathbf{k} & \mathbf{ji} &= -\mathbf{k} \\ \mathbf{jk} &= \mathbf{i} & \mathbf{kj} &= -\mathbf{i} \\ \mathbf{ki} &= \mathbf{j} & \mathbf{ik} &= -\mathbf{j} \end{aligned}$$

2.1.2 Algebraic properties

Addition and subtraction of quaternions are straightforward and can be performed element by element to form a result that is also a quaternion [46]:

for two quaternions: $q_1 = a_1 + b_1\mathbf{i} + c_1\mathbf{j} + d_1\mathbf{k}$ and $q_2 = a_2 + b_2\mathbf{i} + c_2\mathbf{j} + d_2\mathbf{k}$ we have:

$$q_1 \pm q_2 = (a_1 \pm a_2) + (b_1 \pm b_2)\mathbf{i} + (c_1 \pm c_2)\mathbf{j} + (d_1 \pm d_2)\mathbf{k}$$

or in vector form for $q_1 = [s_1, \mathbf{v}_1]$, $q_2 = [s_2, \mathbf{v}_2]$ we have:

$$q_1 \pm q_2 = [s_1 \pm s_2, \mathbf{v}_1 \pm \mathbf{v}_2]$$

with s scalar and **v** vector part. The multiplication of two quaternions is defined as:

$$\begin{aligned} q_1 q_2 := q_1 \otimes q_2 :=& (a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2) + \\ & (a_1 b_2 + b_1 a_2 + c_1 d_2 + d_1 c_2)\mathbf{i} + \\ & (a_1 c_2 + b_1 d_2 + c_1 a_2 + d_1 b_2)\mathbf{j} + \\ & (a_1 d_2 + b_1 c_2 + c_1 b_2 + d_1 a_2)\mathbf{k} \end{aligned}$$

or in a vector form:

$$\begin{aligned} q_1 \otimes q_2 &= [s_1, \mathbf{v}_1] \otimes [s_2, \mathbf{v}_2] \\ &= [s_1 s_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2] \end{aligned}$$

where $(\cdot)$ denotes the dot product and $(\times)$ the cross product of vectors $\mathbf{v}_1$ and $\mathbf{v}_2$. Quaternion multiplication is associative and distributive.

2.1.3 Other properties

The conjugate of a quaternion is defined as [46]:

$$q^* := a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k}$$

or in vector form:

$$q = [s, -\mathbf{v}].$$

The norm of a quaternion is defined as:

$$\|q\| := \sqrt{qq^*} = \sqrt{q^* q} = \sqrt{a^2 + b^2 + c^2 + d^2}.$$

If the norm of a quaternion equals to 1 then it is called a unit norm quaternion. This property is important for the representation and calculations of a rotation of a vector as a unit norm quaternion rotates a vector around its quaternion vector part.

The multiplicative inverse element of a quaternion q or the reciprocal is then:

$$q^{-1} = \frac{q^*}{\|q\|^2}$$

and for unit norm quaternions it holds:

$$q^{-1} = q^*.$$

2.2 Dual numbers

Dual numbers were introduced in 1873 by William Clifford and they are an extension of the real numbers. They were first used to represent the dual angle which measures the relative position of two skew lines in space [48],[49],[50].

2.2.1 Definition

A dual number has the following form [51]:

$$z = a + \varepsilon b$$

where a, b can be real numbers, vectors or matrices. a is called the real part and b is the dual part. ε is a mathematical element (nilpotent) with the important property $\varepsilon^2 = 0$ and $\varepsilon \neq 0$.

2.2.2 Algebraic properties

The addition and subtraction is performed element wise and analogeously to the real numbers [51]. For two dual numbers $z_1 = a_1 + \varepsilon b_1$ and $z_2 = a_2 + \varepsilon b_2$ we have:

$$z_1 \pm z_2 := (a_1 \pm a_2) + \varepsilon(b_1 \pm b_2).$$

The interesting feature is the multiplication which is defined as:

$$z_1 z_2 := (a_1 + \varepsilon b_1)(a_2 + \varepsilon b_2) = a_1 a_2 + \varepsilon(a_1 b_2 + b_1 a_2).$$

The above operations are associative and commutative. The conjugate of a dual number is defined as:

$$z^* := a - \varepsilon b.$$

The inverse of a dual number thus is given by:

$$z^{-1} = a^{-1} - \varepsilon b a^{-2}.$$

2.2.3 Dual matrices

It is important to present the properties of the dual matrices because the nature of the filter that is implemented is based on these features. So for two dual square matrices $A = [a_{ij}]$ and $B = [b_{ij}]$ it holds [51],[49]:

$$A = [R(a_{ij}) + D(a_{ij})] = R(A) + \varepsilon D(A)$$

and

$$B = [R(b_{ij}) + D(b_{ij})] = R(B) + \varepsilon D(B)$$

where R and D are the real an dual part respectively.

Then the following operations hold true [51]:

Addition/subtraction:

$$A \pm B = R(A) \pm R(B) + \varepsilon(D(A) \pm D(B))$$

Multiplication:

$$AB = R(A)R(B) + \varepsilon(R(A)D(B) + D(A)R(B))$$

Multiplication with another dual number:

$$aA = R(a)R(A) + \varepsilon(R(a)D(A) + D(a)R(A))$$

An important property is the inverse of a dual matrix which is given as:

$$A^{-1} = R(A)^{-1} - \varepsilon(R(A)^{-1}D(A)R(A)^{-1})$$

This property holds true only for invertible square matrices.

2.3 Dual quaternions

Dual quaternions are dual numbers that consist of two single quaternions, one for the real part and one for the dual part. Each dual quaternion consists of eight elements [45],[52],[53],[54],[55].

2.3.1 Definition

In general, a dual quaternion is defined as:

$$\hat{q} = q_r + \varepsilon q_d$$

where q_r , q_d are single quaternions and ε the dual unit with the well known properties mentioned above.

2.3.2 Algebraic properties

The following algebraic properties hold for two dual quaternions $\hat{a} = a_r + \varepsilon a_d$ and $\hat{b} = b_r + \varepsilon b_d$ with a_r, a_d, b_r, b_d single quaternions [53]:

Addition/subtraction:

$$\hat{a} + \hat{b} = (a_r + b_r) + \varepsilon(a_d + b_d)$$

Multiplication by a scalar number:

$$\lambda\hat{a} = \lambda a_r + \varepsilon(\lambda a_d)$$

Multiplication:

$$\hat{a}\hat{b} = (a_r b_r) + \varepsilon(a_r b_d + a_d b_r)$$

Conjugation:

$$\hat{a}^* = a_r^* + \varepsilon a_d^*$$

Unit dual quaternion:

$$\hat{a}\hat{a}^* = \hat{a}^*\hat{a} = 1$$

Inverse:

$$\hat{a}^{-1} = a_r^{-1} - \varepsilon(a_r^{-1}a_d a_r^{-1})$$

2.3.3 Pose representation

The mathematical formalization of dual quaternions is able to represent a 6 DOF object transformation, coupling the rotation and translation components in a unit dual quaternion which is defined as [38]:

$$\hat{q} = q_r + \varepsilon q_t \tag{2.1}$$

where $q_r = [q_{0r}, q_{1r}, q_{2r}, q_{3r}]$ is the unit quaternion that represents the rotation component and $q_t = \frac{1}{2}(t \otimes q_r)$ is the quaternion that represents the sum of linear translation and weighted factor of translation from rotation. The vector quaternion $t = [0, t_x, t_y, t_z]$ represents the translation vector.

We can represent only the rotation as a dual quaternion with zero dual part. The translation can be represented in dual quaternion with real part equal to $[1, 0, 0, 0]$ and the dual part equal to $[0, \frac{t_x}{2}, \frac{t_y}{2}, \frac{t_z}{2}]$.

2.4 Kalman filter

The concept behind the Kalman filter is quite old. R. E. Kalman introduced a recursive solution to the discrete-data linear filtering problem in 1960 [56],[8]. Since that time, this technique is very broadly used in applications for navigation, guidance and robotic motion planning. In the next paragraphs a brief introduction of the main features of the Kalman filter is shown. Furthermore, the extended version of the Kalman filter is discussed which is the tool that we used for this project.

2.4.1 Linear Kalman filter

In its core, the Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) way to estimate the state of a process, such that the mean of the squared errors is minimized. The filter estimates are exact conditional probability estimates if all errors are Gaussian [57].

It estimates a process by using a form of feedback control: The filter estimates the process state at some time and then obtains feedback in form of (noisy) measurements. As such, the equations for the Kalman filter fall into two groups: time update equations and measurement update equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimate to obtain an a priori estimate for the next time step.

The measurement update equations are responsible for the feedback which means incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate. The time update equations can also be thought of as a set of predictor equations, while the measurement update equations can be thought of as corrector equations. Indeed, the final estimation algorithm resembles that of a predictor-corrector algorithm for solving numerical problems.

In Fig. 2.1, an overview of the whole operation of the Kalman filter is shown with the according equations for prediction and correction:

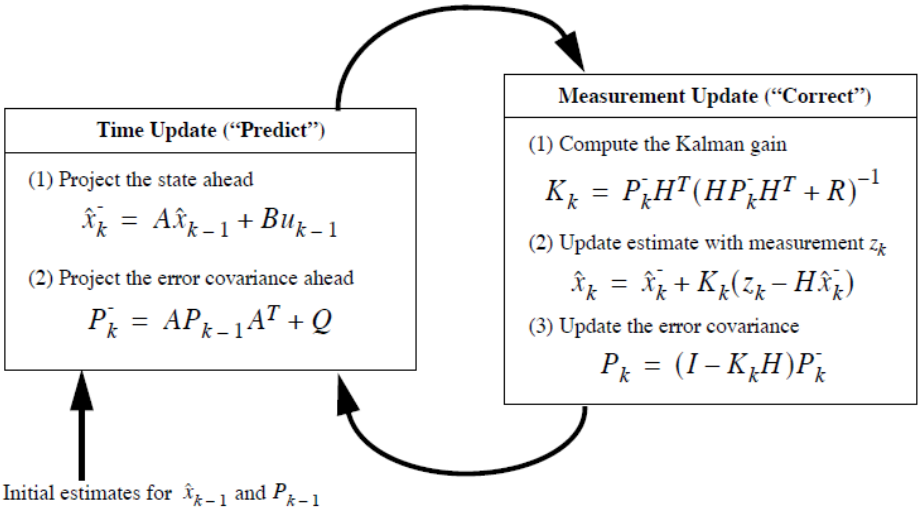


FIGURE 2.1: A complete picture of the operations of the Kalman filter with the according equations for prediction and correction [8]

where:

$\hat{x}_k^-$	predicted (a priori) state estimate at step k
A	the state transition model
$\hat{x}_{k-1}$	corrected (a posteriory) state estimate at step k-1
B	control input model
u_{k-1}	optional control input
P_k^-	predicted (a priori) estimate error covariance at step k
P_{k-1}	corrected (a posteriori) estimate error covariance at step k-1
Q	process noise covariance
K_k	Kalman gain or blending factor at step k
H	the observation model
R	measurement noise covariance
$\hat{x}_k$	corrected (a posteriory) state estimate at step k
z_k	measurement at step k
P_k	corrected (a posteriori) estimate error covariance at step k
I	the identity matrix

At this point we should mention that this is the general form of the equations of the Linear Kalman Filter. In our implementation we don't use any control input so u_{k-1} and B are not considered in the equations.

2.4.2 Extended Kalman filter

The extended form of the linear Kalman filter (EKF) was proposed because the original filter was not able to deal with non-linear systems. So, with this version of the Kalman filter we can exceed this limitation and make use of it in more complex, non-linear systems [8],[58],[59].

This is the reason why we chose the extended Kalman filter to estimate the orientation and translation of the tracked object(s). The EKF is widely used for over 30 years [60],[61],[62] in various mathematical formulations and for many applications including sensor fusion [63],[64]. There are also many more variations of the original Kalman filter for different problems and applications [65].

The main difference of the EKF regarding the functionality of the Kalman filter is that it linearizes the non-linear process at every time instant by computing the Jacobian of the transition matrix about the state vector.

The equations for prediction and correction are the following:

Prediction:

$$\hat{x}_k^- = f(\hat{x}_{k-1}) \qquad \text{(no control input)}$$

$$P_k = A_k P_{k-1} A_k^T + Q$$

Correction:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

$$P_k = (I - K_k H) P_k^-$$

where:

f	state transition function
A_k	the jacobian of the state transition function

The above equations follow the same order as in the Linear Kalman Filter case and the filter works in a recursive way. It is important to state that for our model the process and the measurement noise covariance does not change so the error matrices are constant.

Chapter 3

Sensor fusion model

In this chapter we are going to describe the prototype explicitly, pointing out all the variables, the mathematical equations for the pose representation as well as the extended Kalman filter formulations that are used for the fusion. The data that we use for the fusion are from IMU and vision-based sensors.

The prototype is implemented so that data from up to 4 sensors can be combined to give us the best results. The used method is not restricted to 4 sensors and can be extended to n sensors ($n \geqslant 2$). Furthermore the case in which we have no information from the vision sensor is mentioned as the "occlusion" case and this is included in the prototype.

The basic equations and idea of the dual-quaternion EKF is from Varghese et al.(2015)[38]. We took one step further and implement a filter that can fuse information up to 4 and the possibility of extension to n sensors and also the loss of line of sight is covered.

In the following sections we present firstly the simplest case of fusion of two sensors, one inertial and one vision-based, and then show the extension to a sensor fusion method with multiple sensors.

3.1 Variables

From the involved sensors in order to perform a fusion we need the following data:

From IMU sensor:

q_ω rotation quaternion

ω angular velocity

u_s linear velocity

t_s translation

From vision-based sensor:

q_r rotation quaternion

u_i linear velocity

t_i translation

These variables are going to form the dual quaternions in order be inserted as state variables of the system. The data from the vision-based data can be acquired by mono [66] or stereo optical tracking system [67], with markers[31] or marker-less methods [1], or with SLAM approaches [68]. Every vision-based system or method that can estimate the position and the orientation of an object, can be used by the fusion filter that we introduce in the current study. Furthermore the variables from the inertial sensor(s) are straightforward acquired and inserted into the system for further process [17].

3.2 Dual-quaternion representation

As mentioned in subsection 2.3.3, we use equation 2.1 to represent the pose of the object that is tracked. So the unit dual quaternion for the pose is defined as:

$$\begin{aligned}\hat{q} &= q_r + \varepsilon q_t \\ &= (q_{0r} + q_{1r}\mathbf{i} + q_{2r}\mathbf{j} + q_{3r}\mathbf{k}) + \\ &\quad \varepsilon(\frac{1}{2}(0 + t_x\mathbf{i} + t_y\mathbf{j} + t_z\mathbf{k}) \otimes (q_{0r} + q_{1r}\mathbf{i} + q_{2r}\mathbf{j} + q_{3r}\mathbf{k})) \\ &= (q_{0r} + q_{1r}\mathbf{i} + q_{2r}\mathbf{j} + q_{3r}\mathbf{k}) + \varepsilon(q_{0t} + q_{1t}\mathbf{i} + q_{2t}\mathbf{j} + q_{3t}\mathbf{k}).\end{aligned}$$

Analogously, the unit quaternion that represent the velocity include two single quaternions, one for the angular velocity and a second for the linear one. This

velocity unit dual quaternion is defined as:

$$\begin{aligned}\hat{\omega} &= q_\omega + \varepsilon q_u \\ &= (0 + \omega_x\mathbf{i} + \omega_y\mathbf{j} + \omega_z\mathbf{k}) + \\ &\quad \varepsilon(\frac{1}{2}(0 + u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k}) \otimes (0 + \omega_x\mathbf{i} + \omega_y\mathbf{j} + \omega_z\mathbf{k})) \\ &= (0 + \omega_x\mathbf{i} + \omega_y\mathbf{j} + \omega_z\mathbf{k}) + \varepsilon(q_{0u} + q_{1u}\mathbf{i} + q_{2u}\mathbf{j} + q_{3u}\mathbf{k})\end{aligned}$$

We can see from the above equations that the dual quaternion formulation is able to couple the translation from rotation along with the linear translation and the velocity from angular velocity plus the linear velocity. Every 3D vector (translation, linear velocity, angular velocity) is transformed in a vector quaternion so that the calculations between the quaternions can be made.

3.3 Process model

In the following equations of this chapter, we use the hat ($\hat{}$) notation to indicate that the according variable is in dual form. It can be dual number, vector, quaternion or matrix.

First we define the state vector as:

$$\hat{X} = [\hat{\omega_0}, \hat{\omega_1}, \hat{\omega_2}, \hat{\omega_3}, \hat{q_0}, \hat{q_1}, \hat{q_2}, \hat{q_3}]^T \tag{3.1}$$

where each parameter is represented by a dual quaternion consisting of two simple ones defined as:

$$\begin{aligned}\hat{\omega_0} &= 0 + \varepsilon q_{0u} & \hat{q_0} &= q_{0r} + \varepsilon q_{0t} \\ \hat{\omega_1} &= \omega_x + \varepsilon q_{1u} & \hat{q_1} &= q_{1r} + \varepsilon q_{1t} \\ \hat{\omega_2} &= \omega_y + \varepsilon q_{2u} & \hat{q_2} &= q_{2r} + \varepsilon q_{2t} \\ \hat{\omega_3} &= \omega_z + \varepsilon q_{3u} & \hat{q_3} &= q_{3r} + \varepsilon q_{3t}\end{aligned}$$

We define the measurement vector as:

$$\hat{Z} = [\hat{\omega_0}, \hat{\omega_1}, \hat{\omega_2}, \hat{\omega_3}, \hat{q_{0_I}}, \hat{q_{1_I}}, \hat{q_{2_I}}, \hat{q_{3_I}}, \hat{q_{0_V}}, \hat{q_{1_V}}, \hat{q_{2_V}}, \hat{q_{3_V}}]^T \tag{3.2}$$

The subscripts I and V are used to indicate the measurements from IMU and vision accordingly.

The system is modelled by two mathematical equations that are used for the prediction of the pose and velocity in terms of dual quaternions:

$$\dot{\hat{q}} = \frac{1}{2}\hat{q}\hat{\omega} = [q_r \otimes q_\omega + \varepsilon(q_r \otimes q_u + q_t \otimes q_\omega)] \tag{3.3}$$

$$\dot{\hat{\omega}} = \frac{-1}{\tau}\hat{\omega} \tag{3.4}$$

The dot above $\hat{q}$ and $\hat{\omega}$ indicates the time derivative, which means that the next pose and velocity dual quaternions can be predicted with this. The constant τ is the constant time of the process model. These equations describe the kinematic of the rigid body and are the core for the implementation of the EKF, for which the complete equations are described in the next section.

3.4 EKF equations

The first step is to evaluate equations 3.3 and 3.4. So we get the following equations for the whole system:

$$\dot{\hat{q}_0} = \frac{1}{2}[\hat{q}_0\hat{\omega}_0 - \hat{q}_1\hat{\omega}_1 - \hat{q}_2\hat{\omega}_2 - \hat{q}_3\hat{\omega}_3] \tag{3.5}$$

$$\dot{\hat{q}_1} = \frac{1}{2}[\hat{q}_1\hat{\omega}_0 + \hat{q}_0\hat{\omega}_1 - \hat{q}_3\hat{\omega}_2 + \hat{q}_2\hat{\omega}_3] \tag{3.6}$$

$$\dot{\hat{q}_2} = \frac{1}{2}[\hat{q}_2\hat{\omega}_0 + \hat{q}_3\hat{\omega}_1 + \hat{q}_0\hat{\omega}_2 - \hat{q}_1\hat{\omega}_3] \tag{3.7}$$

$$\dot{\hat{q}_3} = \frac{1}{2}[\hat{q}_3\hat{\omega}_0 - \hat{q}_2\hat{\omega}_1 - \hat{q}_1\hat{\omega}_2 + \hat{q}_0\hat{\omega}_3] \tag{3.8}$$

$$\dot{\hat{\omega}_i} = \frac{-1}{\tau}\hat{\omega}_i \qquad \text{where} \quad i = 0, 1, 2, 3 \tag{3.9}$$

This is how we linarize the non-linear process at every time instant by calculating essentially the Jacobian of the transition matrix.

Afterwards we calculate the difference from the previous to the next state by:

$$\Delta\hat{X}_k^- = A\Delta\hat{X}_{k-1}^- \tag{3.10}$$

with:

$$A = \frac{1}{2} \cdot \begin{bmatrix} 2e^{\frac{-\Delta t}{\tau}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2e^{\frac{-\Delta t}{\tau}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2e^{\frac{-\Delta t}{\tau}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2e^{\frac{-\Delta t}{\tau}} & 0 & 0 & 0 & 0 \\ \widehat{q_0}\Delta t & -\widehat{q_1}\Delta t & -\widehat{q_2}\Delta t & -\widehat{q_3}\Delta t & \widehat{\omega}_0\Delta t & -\widehat{\omega}_1\Delta t & -\widehat{\omega}_2\Delta t & -\widehat{\omega}_3\Delta t \\ \widehat{q_1}\Delta t & \widehat{q_0}\Delta t & -\widehat{q_3}\Delta t & \widehat{q_2}\Delta t & \widehat{\omega}_1\Delta t & \widehat{\omega}_0\Delta t & \widehat{\omega}_3\Delta t & -\widehat{\omega}_2\Delta t \\ \widehat{q_2}\Delta t & \widehat{q_3}\Delta t & \widehat{q_0}\Delta t & -\widehat{q_1}\Delta t & \widehat{\omega}_2\Delta t & -\widehat{\omega}_3\Delta t & \widehat{\omega}_0\Delta t & \widehat{\omega}_1\Delta t \\ \widehat{q_3}\Delta t & -\widehat{q_2}\Delta t & \widehat{q_1}\Delta t & \widehat{q_0}\Delta t & \widehat{\omega}_3\Delta t & \widehat{\omega}_2\Delta t & -\widehat{\omega}_1\Delta t & \widehat{\omega}_0\Delta t \end{bmatrix} \tag{3.11}$$

with Δt the sample time interval.

We acquire the *a priori* estimation $\hat{X}_k^-$ for the parameters of the system by the following equation.

$$\hat{X}_k^- = \hat{X}_{k-1} + \Delta\hat{X}_k^- \tag{3.12}$$

With the above equation we map the state vector into measurement space such that they are comparable and have the same size at step k:

$$\hat{Z}_k := H\hat{X}_k^- \tag{3.13}$$

where H is the observation matrix for the mapping. For the case of two sensors it is defined as:

$$H = \begin{bmatrix} I_4 & 0_{4\times4} \\ 0_{4\times4} & I_4 \\ 0_{4\times4} & I_4 \end{bmatrix}. \tag{3.14}$$

We define the measurement covariance matrix as:

$$R = \begin{bmatrix} I_4 \cdot \sigma_\omega^2 & 0_{4\times4} & 0_{4\times4} \\ 0_{4\times4} & I_4 \cdot \sigma_{q_I}^2 & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} & I_4 \cdot \sigma_{q_V}^2 \end{bmatrix} \tag{3.15}$$

where σ_{ω}^2 , $\sigma_{q_I}^2$ and $\sigma_{q_V}^2$ are the variances that correspond to the angular velocity and pose measurements from IMU and vision-based sensor respectively.

Also we define the process covariance matrix as:

$$Q = \begin{bmatrix} I_4 \cdot p_u & 0_{4\times4} \\ 0_{4\times4} & I_4 \cdot p_q \end{bmatrix} \tag{3.16}$$

where p_u , p_q are the variance associated with velocity and orientation states respectively.

The *a priori* estimation error covariance matrix is given by:

$$\hat{P}_k^- = \hat{A}\hat{P}_{k-1}^-\hat{A}^T + Q \tag{3.17}$$

The Kalman gain is defined as:

$$\hat{K} = \hat{P}_k^- H^T (H\hat{P}_k^- H^T + R)^{-1} \tag{3.18}$$

The *a posteriori* estimation error covariance matrix is defined as:

$$\hat{P}_k = (I - \hat{K}_k H)\hat{P}_k^- \tag{3.19}$$

Finally the estimated state after the correction is given by:

$$\hat{X}_k = \hat{X}_k^- + \hat{K}_k(\hat{Z}_k - H\hat{X}_k^-) \tag{3.20}$$

The EKF runs recursively which means that the next state is updated by the next measurement and the previous estimation according to their error covariance which is indicated every step by the Kalman gain.

As mentioned before these equations are used in the case that we have information from one IMU and one vision sensor. In the next section we describe a possible extension of the EKF implementation for n sensors and the explicit version for up to four sensors.

3.5 Extension to multiple sensors

For the concept of the fusion of more than two sensors, the main implementation of the filter remains the same but there are some additional parameters and changes in order to include the additional information.

The parameters that are essentially changing are firstly the ones in the measurement vector (eq.3.2). In general whenever we have an additional IMU sensor, eight more parameters are inserted in the measurement vector. On the other hand when we use an additional vision-based sensor only four parameters will be added to the measurement vector. This is defined independent of the formulation of the basic equations of the Kalman filter.

Along with the additional information from the measurements what is changing inside the Kalman filter are the σ_{ω}^2 , $\sigma_{q_I}^2$ and $\sigma_{q_V}^2$ which are referred to the error variances of the angular velocity, the IMU and the vision measurements respectively. This variables are connected directly to the measurement covariance matrix R (eq.3.15) which changes accordingly.

The last thing that inevitably alters is the observation matrix H (eq.3.14) which is responsible for the mapping of the measurements and has to include the new parameters of the new sensors.

In order to show an example, we explicitly look at three cases apart from the first one that we have already described and also include the occlusion case. Finally we show the general case for n sensors.

3.5.1 Case 1 : 1 IMU and 1 Vision Sensor

For consistency we will refer in this paragraph the equations from the previous section that were used for the case that we have data from one IMU and one vision-based sensor.

The measurement vector is given by equation 3.2. It consist of 12 dual quaternions, 8 of them come from the IMU and 4 from the vision-based sensor.

The observation matrix H which maps the state vector variables to the measurement vector is defined by equation 3.14.

Finally the covariance matrix for the measurement error is given by equation 3.15.

3.5.2 Case 2 : 2 IMUs and 1 Vision Sensor

The measurement vector is defined as:

$$\begin{aligned}\hat{Z} =&[\hat{\omega_0}^1, \hat{\omega_1}^1, \hat{\omega_2}^1, \hat{\omega_3}^1,\\ &\hat{\omega_0}^2, \hat{\omega_1}^2, \hat{\omega_2}^2, \hat{\omega_3}^2,\\ &\hat{q_{0_I}}^1, \hat{q_{1_I}}^1, \hat{q_{2_I}}^1, \hat{q_{3_I}}^1,\\ &\hat{q_{0_I}}^2, \hat{q_{1_I}}^2, \hat{q_{2_I}}^2, \hat{q_{3_I}}^2,\\ &\hat{q_{0_V}}, \hat{q_{1_V}}, \hat{q_{2_V}}, \hat{q_{3_V}}]^T\end{aligned} \tag{3.21}$$

where the superscripts 1 and 2 indicate the dual quaternion of velocity and pose from the first and second IMU sensor.

The observation matrix takes the form:

$$H = \begin{bmatrix} I_4 & 0_{4\times4} \\ I_4 & 0_{4\times4} \\ 0_{4\times4} & I_4 \\ 0_{4\times4} & I_4 \\ 0_{4\times4} & I_4 \end{bmatrix} \tag{3.22}$$

so that the state vector can be mapped into the according measurements.

The measurement covariance matrix is defined as:

$$R = \begin{bmatrix} I_4 \cdot \sigma^2_{\omega_1} & 0_{4\times4} & 0_{4\times4} & 0_{4\times4} & 0_{4\times4} \\ 0_{4\times4} & I_4 \cdot \sigma^2_{\omega_2} & 0_{4\times4} & 0_{4\times4} & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} & I_4 \cdot \sigma^2_{q_{I_1}} & 0_{4\times4} & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} & 0_{4\times4} & I_4 \cdot \sigma^2_{q_{I_2}} & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} & 0_{4\times4} & 0_{4\times4} & I_4 \cdot \sigma^2_{q_V} \end{bmatrix} \tag{3.23}$$

which is becoming a 20×20 matrix with $\sigma^2_{\omega_2}$ and $\sigma^2_{q_{I_2}}$ the corresponding variances of the error for angular velocity and pose from the additional IMU sensor.

3.5.3 Case 3 : 1 IMU and 2 Vision Sensors

For this case the size of the matrices alters accordingly to the extra parameters that are introduced from the additional vision sensor. As mentioned above we add only four extra parameters to the measurement vector. Thus it is defined as:

$$\begin{aligned}\hat{Z} =&[\hat{\omega_0}, \hat{\omega_1}, \hat{\omega_2}, \hat{\omega_3},\\ &\hat{q_{0_I}}, \hat{q_{1_I}}, \hat{q_{2_I}}, \hat{q_{3_I}},\\ &\hat{q_{0_V}}^1, \hat{q_{1_V}}^1, \hat{q_{2_V}}^1, \hat{q_{3_V}}^1,\\ &\hat{q_{0_V}}^2, \hat{q_{1_V}}^2, \hat{q_{2_V}}^2, \hat{q_{3_V}}^2]^T\end{aligned} \tag{3.24}$$

where the superscripts are indicating the first and second vision based sensor that is added.

The observation matrix takes the form:

$$H = \begin{bmatrix} I_4 & 0_{4\times4} \\ 0_{4\times4} & I_4 \\ 0_{4\times4} & I_4 \\ 0_{4\times4} & I_4 \end{bmatrix}. \tag{3.25}$$

The measurement covariance matrix R is defined as:

$$R = \begin{bmatrix} I_4 \cdot \sigma_\omega^2 & 0_{4\times4} & 0_{4\times4} & 0_{4\times4} \\ 0_{4\times4} & I_4 \cdot \sigma_{q_I}^2 & 0_{4\times4} & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} & I_4 \cdot \sigma_{q_{V_1}}^2 & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} & 0_{4\times4} & I_4 \cdot \sigma_{q_{V_2}}^2 \end{bmatrix} \tag{3.26}$$

which is becoming a 16×16 matrix with $\sigma_{q_{V_2}}^2$ the corresponding variance of the added vision sensor.

3.5.4 Case 4 : 2 IMUs and 2 Vision Sensors

For the case of two additional sensors, one of each kind we are inserting 12 extra parameters to the measurement vector such that it has the form:

$$\begin{aligned}\hat{Z} =&[\hat{\omega_0}^1, \hat{\omega_1}^1, \hat{\omega_2}^1, \hat{\omega_3}^1,\\ &\hat{\omega_0}^2, \hat{\omega_1}^2, \hat{\omega_2}^2, \hat{\omega_3}^2,\\ &\hat{q_{0_I}}^1, \hat{q_{1_I}}^1, \hat{q_{2_I}}^1, \hat{q_{3_I}}^1,\\ &\hat{q_{0_I}}^2, \hat{q_{1_I}}^2, \hat{q_{2_I}}^2, \hat{q_{3_I}}^2,\\ &\hat{q_{0_V}}^1, \hat{q_{1_V}}^1, \hat{q_{2_V}}^1, \hat{q_{3_V}}^1,\\ &\hat{q_{0_V}}^2, \hat{q_{1_V}}^2, \hat{q_{2_V}}^2, \hat{q_{3_V}}^2]^T\end{aligned} \tag{3.27}$$

where the superscripts indicate the additional parameters that are introduced both from the extra IMU (eight parameters) and vision-based sensor (four parameters).

The observation matrix is defined as:

$$H = \begin{bmatrix} I_4 & 0_{4\times4} \\ I_4 & 0_{4\times4} \\ 0_{4\times4} & I_4 \\ 0_{4\times4} & I_4 \\ 0_{4\times4} & I_4 \\ 0_{4\times4} & I_4 \end{bmatrix}. \tag{3.28}$$

Finally the measurement covariance matrix R is 24×24 and is defined as:

$$R = \begin{bmatrix} I_4 \cdot \sigma^2_{\omega_1} & 0_{4\times4} & 0_{4\times4} & 0_{4\times4} & 0_{4\times4} & 0_{4\times4} \\ 0_{4\times4} & I_4 \cdot \sigma^2_{\omega_2} & 0_{4\times4} & 0_{4\times4} & 0_{4\times4} & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} & I_4 \cdot \sigma^2_{q_{I_1}} & 0_{4\times4} & 0_{4\times4} & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} & 0_{4\times4} & I_4 \cdot \sigma^2_{q_{I_2}} & 0_{4\times4} & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} & 0_{4\times4} & 0_{4\times4} & I_4 \cdot \sigma^2_{q_{V_1}} & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} & 0_{4\times4} & 0_{4\times4} & 0_{4\times4} & I_4 \cdot \sigma^2_{q_{V_2}} \end{bmatrix} \tag{3.29}$$

with the according additions for the error covariances from both additional sensors.

3.5.5 Case 5 : Occlusion

For the occlusion case we have two modes, one when there is no occlusion and one when we have no information from the vision-based sensor. The case is implemented for 2 IMUs and one vision sensor.

When we acquire data from all the sensors of the system, the filter acts like the case described in section (3.5.2). As long as the data from the vision sensor is not inserted into the filter we switch to a special case that calculates the system parameters only with the data from the IMUs.

Then the measurement vector becomes:

$$\begin{aligned}\hat{Z} =&[\hat{\omega_0}^1, \hat{\omega_1}^1, \hat{\omega_2}^1, \hat{\omega_3}^1,\\ &\hat{\omega_0}^2, \hat{\omega_1}^2, \hat{\omega_2}^2, \hat{\omega_3}^2,\\ &\hat{q_{0_I}}^1, \hat{q_{1_I}}^1, \hat{q_{2_I}}^1, \hat{q_{3_I}}^1,\\ &\hat{q_{0_I}}^2, \hat{q_{1_I}}^2, \hat{q_{2_I}}^2, \hat{q_{3_I}}^2]^T\end{aligned} \tag{3.30}$$

The observation matrix takes the form:

$$H = \begin{bmatrix} I_4 & 0_{4\times4} \\ I_4 & 0_{4\times4} \\ 0_{4\times4} & I_4 \\ 0_{4\times4} & I_4 \end{bmatrix}. \tag{3.31}$$

Finally the measurement error covariance matrix is defined by:

$$R = \begin{bmatrix} I_4 \cdot \sigma^2_{\omega_1} & 0_{4\times4} & 0_{4\times4} & 0_{4\times4} \\ 0_{4\times4} & I_4 \cdot \sigma^2_{\omega_2} & 0_{4\times4} & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} & I_4 \cdot \sigma^2_{q_{I_1}} & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} & 0_{4\times4} & I_4 \cdot \sigma^2_{q_{I_2}} \end{bmatrix} \tag{3.32}$$

It is well known that the inertial sensors are very sensitive to drift when it comes to the calculation of the velocity and much more for the calculation of the position. This is because we have to integrate the acceleration twice and this way the error propagates quadratically which leads to drift.

Nevertheless for short time occlusions the filter can give valid results which are analyzed in the experiment chapter hereafter.

3.5.6 General case for n sensors

As mentioned in the beginning of the chapter, the fusion method that we use can be extended to n number of sensors. We assume that we have i number of IMU sensors and j number of vision-based sensors.

We generalize the method by defining the measurement vector as:

$$\begin{aligned}\hat{Z} =&[\hat{\omega_0}^1, \hat{\omega_1}^1, \hat{\omega_2}^1, \hat{\omega_3}^1,\\ &\hat{\omega_0}^2, \hat{\omega_1}^2, \hat{\omega_2}^2, \hat{\omega_3}^2,\\ &\dots, \dots, \dots, \dots,\\ &\hat{\omega_0}^i, \hat{\omega_1}^i, \hat{\omega_2}^i, \hat{\omega_3}^i,\\ &\hat{q_{0_I}}^1, \hat{q_{1_I}}^1, \hat{q_{2_I}}^1, \hat{q_{3_I}}^1,\\ &\hat{q_{0_I}}^2, \hat{q_{1_I}}^2, \hat{q_{2_I}}^2, \hat{q_{3_I}}^2,\\ &\dots, \dots, \dots, \dots,\\ &\hat{q_{0_I}}^i, \hat{q_{1_I}}^i, \hat{q_{2_I}}^i, \hat{q_{3_I}}^i,\\ &\hat{q_{0_V}}^1, \hat{q_{1_V}}^1, \hat{q_{2_V}}^1, \hat{q_{3_V}}^1,\\ &\hat{q_{0_V}}^2, \hat{q_{1_V}}^2, \hat{q_{2_V}}^2, \hat{q_{3_V}}^2,\\ &\dots, \dots, \dots, \dots,\\ &\hat{q_{0_V}}^j, \hat{q_{1_V}}^j, \hat{q_{2_V}}^j, \hat{q_{3_V}}^j]^T\end{aligned} \tag{3.33}$$

The observation matrix is generalized as:

$$H = \begin{bmatrix} I_4 & 0_{4\times4} & \\ \vdots & \vdots & i\ lines \\ 0_{4\times4} & I_4 & \\ \vdots & \vdots & i\ lines \\ 0_{4\times4} & I_4 & \\ \vdots & \vdots & j\ lines \end{bmatrix} \tag{3.34}$$

The measurement error covariance matrix is extended the following way:

- If an IMU sensor is added, we add to the diagonal of the basic measurement error covariance matrix (equation 3.15) the according σ_ω^2 and $\sigma_{q_I}^2$ multiplied by the identity 4 by 4 matrix.

- If a vision-based sensor is added, we add to the diagonal of the matrix the according $\sigma^2_{q_V}$ multiplied by the identity 4 by 4 matrix.
- All the other entries will be filled with the $0_{4\times4}$ matrix so that the final matrix is a square matrix.

We can express the matrix equation as:

$$\begin{aligned} R = diag(&I_4 \cdot \sigma^2_{\omega_1}, I_4 \cdot \sigma^2_{\omega_2}, \ldots, I_4 \cdot \sigma^2_{\omega_n}, \\ &I_4 \cdot \sigma^2_{q_{I_1}}, I_4 \cdot \sigma^2_{q_{I_2}}, \ldots, I_4 \cdot \sigma^2_{q_{I_n}}, \\ &I_4 \cdot \sigma^2_{q_{V_1}}, I_4 \cdot \sigma^2_{q_{V_2}}, \ldots, I_4 \cdot \sigma^2_{q_{V_n}}) \end{aligned} \tag{3.35}$$

Chapter 4

Experiments

The experiments are done with synthetic data, data from an existing dataset[9] and real data. In this chapter we provide all the measurements, the parameters and the methods that we used for these three cases.

4.1 Synthetic data

We generate artificial data in order to test the filter for all the individual cases as described in sections 3.4 and 3.5.

For the testing purposes we use 100 poses with the same number of the following components. The sample rate of the artificial poses is $\Delta t = 0.03$ s so the time is considered to be 3 s in total. The constant time of the process model is $\tau = 0.5$ s. We firstly generate the rotation quaternion data, the translation vector, the angular velocity and the linear velocity.

- The range of the rotation angle in the x-axis which is represented by the rotation quaternion varies from 0° to 90°.
- The range of the translation in x-axis which is represented by the translation vector varies from 0 to 100 meters.
- The angular velocity also in x-axis is implied by the rotation and it has the value of 0.5235 rad/s.

- The linear velocity is also implied by the position and time interval and has the value of 33.33 m/s.

For every sensor we add white Gaussian noise in order to evaluate the filter and see how it performs in every case. We assume that the measurements from the OTS are more accurate than those from the IMU so the noise is added accordingly. More specifically:

- For the IMU sensors we set a variance of 0.001 rad^2 for the rotation and a variance of 100 m^2 for the translation.
- For the OTS sensors we set a variance of 0.01 rad^2 for the rotation and a variance of 10 m^2 for the translation.
- For the angular velocity we assume noise with a variance of 0.1 $(rad/s)^2$.
- For the linear velocity we assume higher noise than the one for the angular velocity with a variance of $1(m/s)^2$.

For the EKF initialization we must provide initial conditions for every entry. We initialize the filter entries as follows:

- translation vector $\rightarrow$ (0,0,0)
- rotation quaternion $\rightarrow$ (1,0,0,0)
- linear velocity vector $\rightarrow$ (33.3,0,0)
- angular velocity vector $\rightarrow$ (0.5235,0,0)

We should mention here that the initialization of the entries is on purpose the same as the first measurements in order to be able to and compare the root mean square error (RMSE) and variance of the output with the individual sensors and the ground truth values.

After the initialization, we feed the filter with the artificial sequences and the algorithm works in a recursive way as described in section 2.4.2.

For each case we also define the filter parameters as presented in section 3.5 for the 4 individual cases and also the occlusion case.

- Case 1: 1 IMU and 1 Vision Sensor

Parameter	**Values**
σ^2_{ω}	0.1 rad^2/s^2
$\sigma^2_{q_I}$	0.1
$\sigma^2_{q_V}$	0.01
p_u	0.01
p_q	0.01

TABLE 4.1: Filter tuning parameters for Case 1

- Case 2: 2 IMUs and 1 Vision Sensor

Parameter	**Values**
$\sigma^2_{\omega_1}$	0.1 rad^2/s^2
$\sigma^2_{\omega_2}$	0.1 rad^2/s^2
$\sigma^2_{q_{I_1}}$	0.1
$\sigma^2_{q_{I_2}}$	0.1
$\sigma^2_{q_V}$	0.01
p_u	0.01
p_q	0.01

TABLE 4.2: Filter tuning parameters for Case 2

- Case 3: 1 IMU and 2 Vision Sensors

Parameter	**Values**
σ^2_{ω}	0.1 rad^2/s^2
$\sigma^2_{q_I}$	0.1
$\sigma^2_{q_{V_1}}$	0.01
$\sigma^2_{q_{V_1}}$	0.01
p_u	0.01
p_q	0.01

TABLE 4.3: Filter tuning parameters for Case 3

- Case 4: 2 IMUs and 2 Vision Sensors

Parameter	**Values**
$\sigma^2_{\omega_1}$	0.1 rad^2/s^2
$\sigma^2_{\omega_2}$	0.1 rad^2/s^2
$\sigma^2_{q_{I_1}}$	0.1
$\sigma^2_{q_{I_2}}$	0.1
$\sigma^2_{q_{V_1}}$	0.01
$\sigma^2_{q_{V_2}}$	0.01
p_u	0.01
p_q	0.01

TABLE 4.4: Filter tuning parameters for Case 4

- Case 5: Occlusion

  For this case we consider a hypothetical case of three sensors (2 IMUs and one vision sensor). We define two occasions, one in which we have the data from all three sensors and one in which we loose the 'line of sight' and so the data from the OTS is not fed in the filter. The parameters are shown below.

No Occlusion		Occlusion	
Parameter	**Values**	**Parameter**	**Values**
$\sigma^2_{\omega_1}$	0.1 rad^2/s^2	$\sigma^2_{\omega_1}$	0.1 rad^2/s^2
$\sigma^2_{\omega_2}$	0.1 rad^2/s^2	$\sigma^2_{\omega_2}$	0.1 rad^2/s^2
$\sigma^2_{q_{I_1}}$	0.1	$\sigma^2_{q_{I_1}}$	0.1
$\sigma^2_{q_{I_2}}$	0.1	$\sigma^2_{q_{I_2}}$	0.1
$\sigma^2_{q_V}$	0.01	-	-
p_u	0.01	p_u	0.01
p_q	0.01	p_q	0.01

TABLE 4.5: Filter tuning parameters for Occlusion case

The artificial sequences of the translation vector and the rotation quaternion of the OTS are interrupted for 20 poses and the filter is taking the last pose from the OTS to calculate the next one, which is only depended on the IMU measurements and the processing model. By the end of the 'Occlusion' the filter is working again with all the data from all sensors.

4.2 Dataset

For the application evaluation of our method we used the data from the PennCOSYVIO dataset [9]. It is a collection of synchronized video and IMU data recorded at the University of Pennsylvania's Singh Center in April 2016. The trajectory is about 150 m long and includes change of environment from outdoors to indoors, rapid rotations, changes in lighting, different textures, repetitive structures and large glass surfaces.

Ten sensors are mounted to a hand-held rig: seven cameras and three IMUs in total, including two Google Project Tango tablets, four GoPro Hero 4 Cameras, and a VI (Visual-Inertial) sensor.

A student hold the rig and starting outside the building, he go inside covering some distance indoors and finally he comes out again to finish at the starting point. Below, a figure shows how the sensors are attached to the rig and the orientation of each sensor is shown.

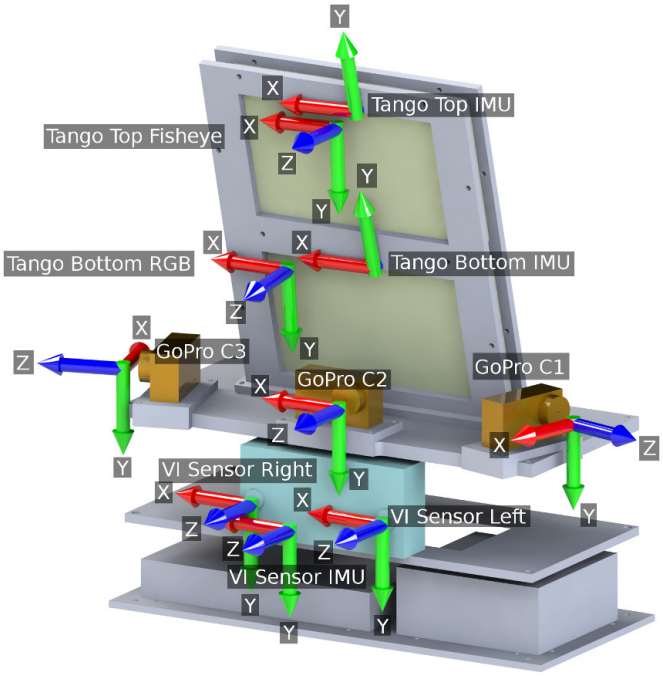


FIGURE 4.1: Sensor rig with orientations of all ten sensors [9]

The characteristics of the sensors are presented briefly:

- C1,C2,C3

  - ⋆ GoPro Hero 4 Black (GoPro Inc., California, U.S.)
  - ⋆ Rolling shutter
  - ⋆ FOV: 69.5° vert., 118.2° horiz.

- VI-Sensor
  - ⋆ Stereo camera: 2 x Aptina MT9V034 (On Semiconductor, Arizona, U.S.)
  - ⋆ Gray 2x752x480 at 20fps (rectified), global shutter
  - ⋆ FOV: 57° vert., 2 x 80° horiz.
  - ⋆ IMU: ADIS16488 (Analog devices Inc.,Massachusetts, U.S.) at 200Hz

- Tango Bottom
  - ⋆ Google Project Tango Yellowstone 7in tablet
  - ⋆ RGB 1920x1080 at 30fps, rolling shutter
  - ⋆ FOV: 31° vert., 52° horiz.
  - ⋆ Accelerometer at 128Hz
  - ⋆ Gyroscope at 100Hz

- Tango Top
  - ⋆ Google Project Tango Yellowstone 7in tablet
  - ⋆ Gray 640x480 at 30fps, global shutter
  - ⋆ FOV: 100° vert., 132° horiz.
  - ⋆ Accelerometer at 128Hz
  - ⋆ Gyroscope at 100Hz

All the data is synchronized. The authors provide two versions of the same route, a slow and a fast one. However the ground truth is provided only for the slow version and this is the one that we used for the evaluation of our filter.

These data are the translation and rotation from the tango bottom sensor and the accelerometer and gyroscope data from the VI sensor.

In order to acquire the rotation quaternions from the VI sensor we used Madwick's filter [35],[69] which is implemented in Matlab by x-io Technologies [70].

It is applicable to inertial measurement units (IMUs) consisting of tri-axis gyroscopes, accelerometers, and magnetic angular rate and gravity (MARG) sensor arrays that also include tri-axis magnetometers. The algorithm uses a quaternion representation, allowing accelerometer and magnetometer data to be used in an analytically derived and optimised gradient descent algorithm to compute the direction of the gyroscope measurement error as a quaternion derivative [35]. For the purposes of the current study we use the implementation that takes as input only the acceleration and gyroscopic data.

We have included two separate cases for using the dataset, one with all the data from both sensors and one by breaking the vision-based data sequence to represent the loss of line of sight.

The parameters for the first subcase are shown below:

Parameter	**Values**
σ_{ω}^2	0.01 rad^2/s^2
$\sigma_{q_I}^2$	0.1
$\sigma_{q_V}^2$	0.01
p_u	0.1
p_q	0.1

TABLE 4.6: Filter tuning parameters for Dataset (no occlusion)

The parameters for the second subcase are shown in the next table:

No Occlusion		Occlusion	
Parameter	**Values**	**Parameter**	**Values**
σ_{ω}^2	0.1 rad^2/s^2	σ_{ω}^2	0.1 rad^2/s^2
$\sigma_{q_I}^2$	0.1	$\sigma_{q_I}^2$	0.1
$\sigma_{q_V}^2$	0.01	-	-
p_u	0.1	p_u	0.1
p_q	0.1	p_q	0.1

TABLE 4.7: Filter tuning parameters for Dataset (Occlusion case)

The sequence is interrupted in the 1000th pose for 100 poses. As the time interval between the poses is 0.03s, the occlusion occurs for 3s. During this time as in the

synthetic data occlusion case the filter is working with the prediction model and the IMU data.

We should mention that during the occlusion the position is calculated by double integration of the corresponding acceleration values. As this method is very inaccurate we choose not to calculate the position for the whole trajectory rather than only when the loss of line of sight occurs, to reduce the drift effect only at this period and have better estimation.

Finally we initialize the entries of the filter with the first measurement values, as we did with the synthetic data:

4.3 Real Data

For further testing of the sensor fusion EKF filter, we have also used real data from experiments that were held in FRAMOS GmbH. The hardware that we use is a monocular camera (GCC2462C, SMARTEK Vision, Croatia) and an IMU sensor (PhidgetSpatial 1044, Phidgets Inc., Canada).

At first the IMU sensor is attached to an object on which we attach also an ArUco marker. ArUco is an OpenSource library for camera pose estimation using squared markers [71]. A sequence of images are acquired by the camera and through the image processing we get the poses for each timestamp with time interval of 0.05 s. At the same time we get also the measurements from the IMU sensor. Except from the angular velocity and the acceleration we acquire the rotation quaternions, the same way we did for the dataset, via the Madwick's filter [35],[69].

All the data are transferred to the fusion filter via OpenIGTLink, an open-source network communication interface [72], implemented in Matlab. The interface allows us to obtain the data from the platform with the according timestamp so that we can further process them. Last step is passing all the data in the EKF fusion filter with the following parameters which are shown in the above matrix.

Parameter	**Values**
σ^2_{ω}	0.1 rad^2/s^2
$\sigma^2_{q_I}$	0.1
$\sigma^2_{q_V}$	0.1
p_u	0.1
p_q	0.1

TABLE 4.8: Filter tuning parameters for Real Data

Chapter 5

Results and evaluation

5.1 Synthetic data results

In this chapter we present the results from the experiments with synthetic data that are described above. We divide it in two sections. The first one contains the results from the first four cases that the data from IMU and vision sensors are fused without loss of line of sight. The second one includes the occlusion case.

5.1.1 Results for Cases 1 - 4

In the figures 5.1 - 5.6 below we show the representative figures for position and rotation in x, y and z axis for Case 1 : 1 IMU and 1 Vision Sensor.

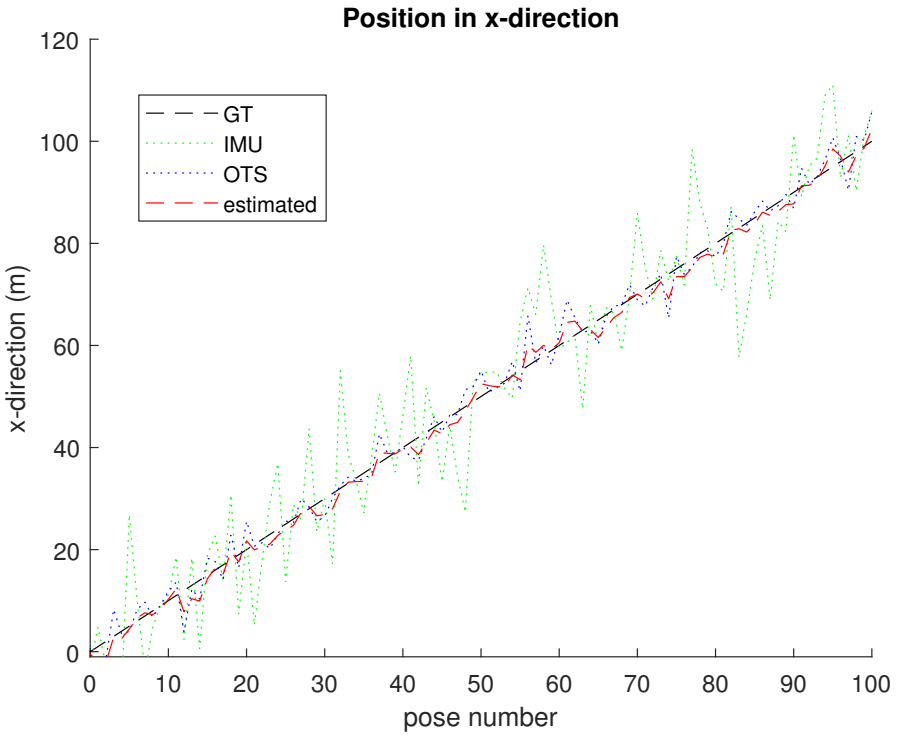


FIGURE 5.1: Case 1, position in x-direction

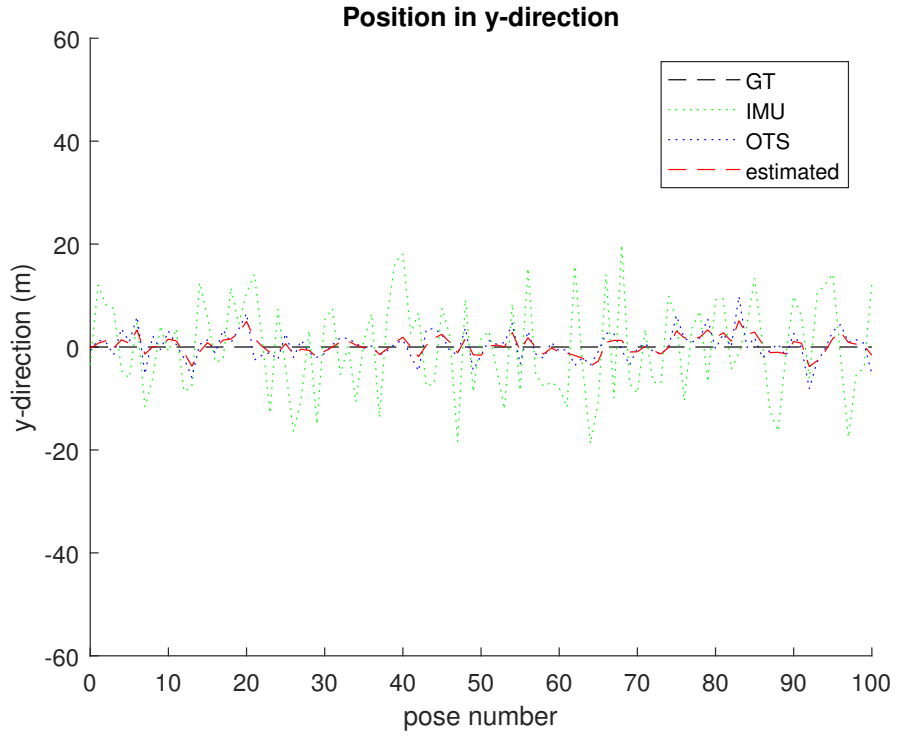


FIGURE 5.2: Case 1, position in y-direction

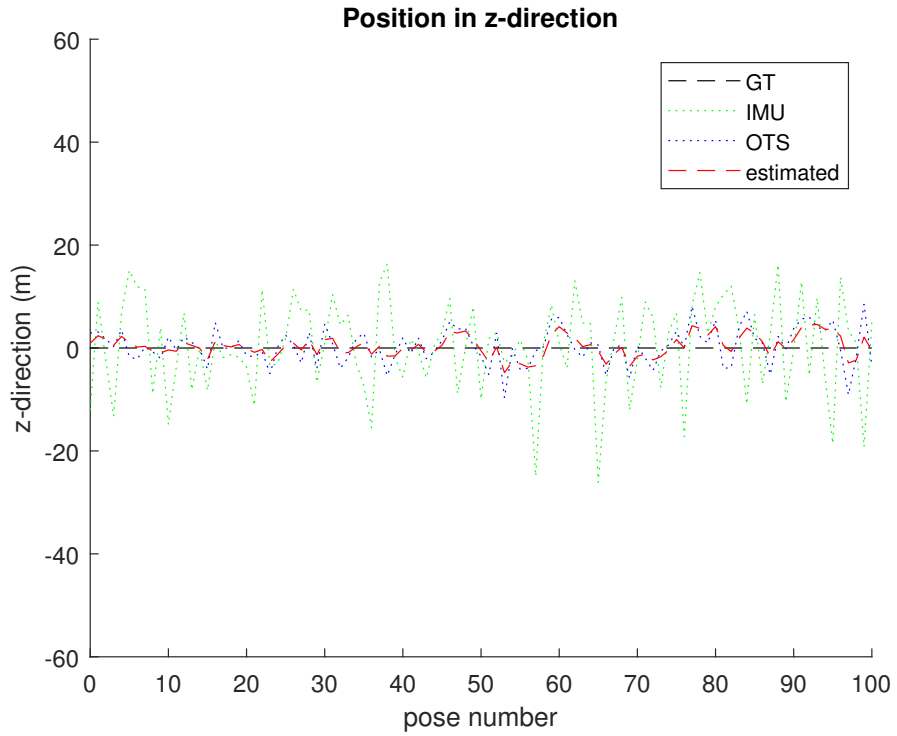


FIGURE 5.3: Case 1, position in z-direction

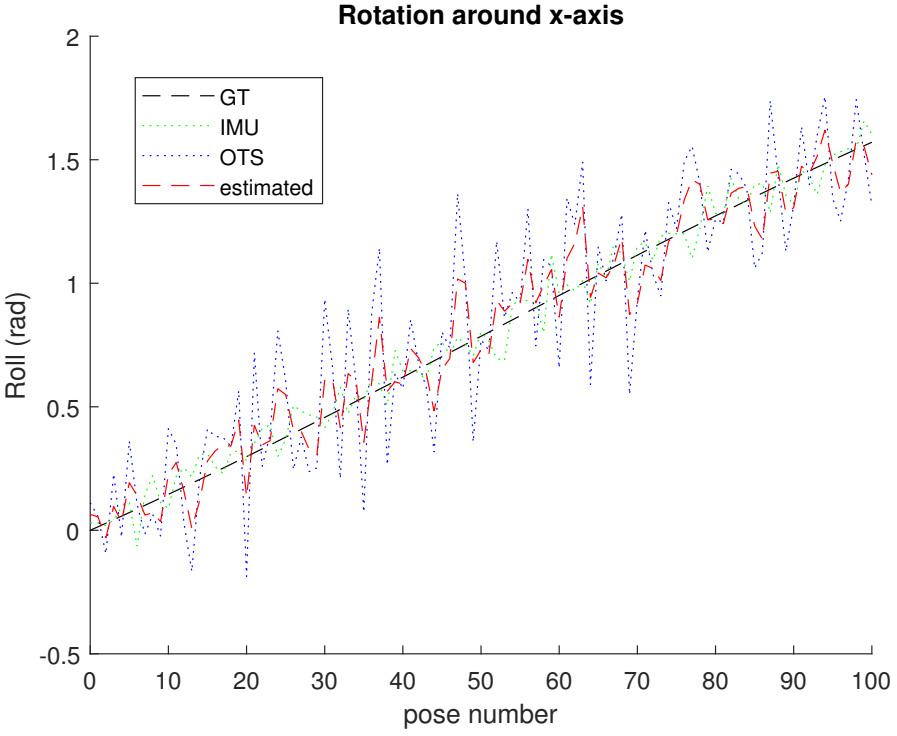


FIGURE 5.4: Case 1, rotation around x-axis

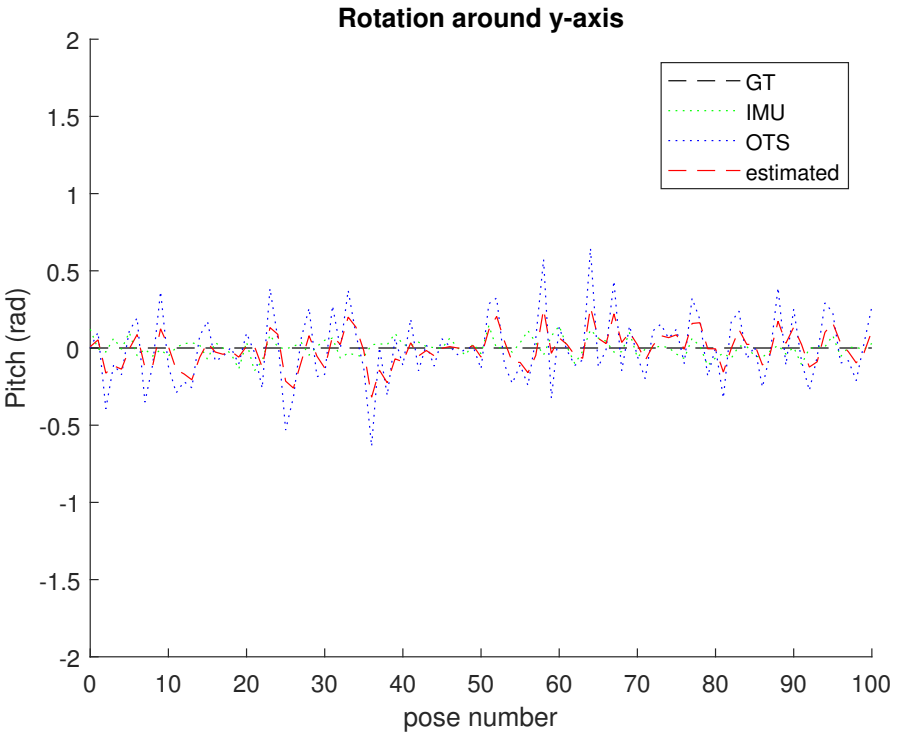


FIGURE 5.5: Case 1, rotation around y-axis

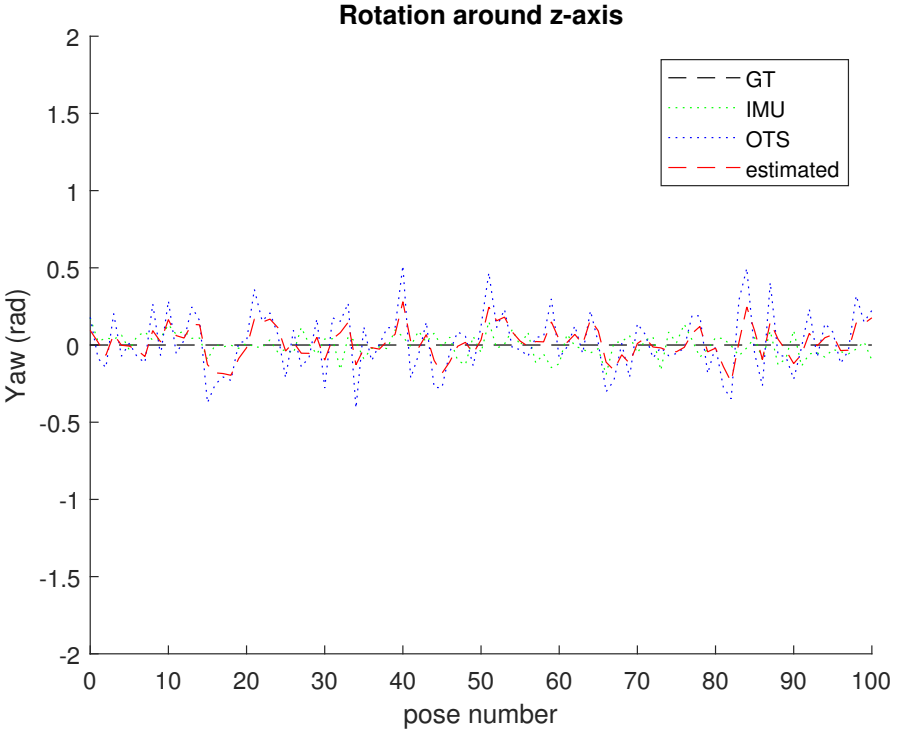


FIGURE 5.6: Case 1, rotation around z-axis

In the following tables we give a quantitative error analysis for the first case, presenting the corresponding mean, median and standard deviation (std) of the error of each sensor and the fused method for position and rotation.

We calculate the position error as the square root of the norm of the difference between the measurement and the ground truth vector according to the following formula:

$$\text{position error} = \sqrt{\|\mathbf{v} - \mathbf{u}\|}$$

where v is the measurement vector and u is the ground truth vector.

The rotation angle error is considered the error of the angle between the measurement and the ground truth quaternion. It is shown in the following equation:

$$\text{angle error} = arccos(2 \cdot (\mathbf{q_v} \cdot \mathbf{q_v})^2 - 1)$$

where q_v is the measurement and q_u is the ground truth quaternion.

	Position error		
	mean (m)	**median** (m)	**std** (m)
Vision	2.1480	2.1946	0.5348
IMU	3.8347	3.8244	0.7338
Fusion	1.6877	1.7131	0.4074

TABLE 5.1: Case 1: mean, median and standard deviation for position error for sensors and fusion

	Rotation angle error		
	mean (°)	**median** (°)	**std** (°)
Vision	20.1310	18.4702	8.6150
IMU	5.8394	5.1448	2.7851
Fusion	10.1581	9.9927	4.0125

TABLE 5.2: Case 1: mean, median and standard deviation for rotation angle error for sensors and fusion

From the above figures for the first case and also from the tables we make two major observations. Firstly we may see that both for the translation and the rotation

components, the estimated values are better than those from the vision-based sensor. This is obvious in the graphs but also in the error values as we have for the position a mean error of 2.1480 m for the vision with an standard deviation of 0.5348 m while for the fusion we have 1.6877 m and 0.4074 m accordingly. Regarding the angle error is also reduced from a mean value of 20.1310° with standard deviation of 8.6150°, to the corresponding values for the fusion which are 10.1581° and 4.0125° accordingly.

The second observation is that for the position we have more accurate results from the fusion method than the other two separate sensors, but for the rotation the error for the IMU sensor is lower than the one from the fusion method. For the IMU the mean error is 5.8394° with standard deviation of 2.7851°, almost half of the fusion corresponding values and this is because we rely more on the vision sensor measurements and as they have more noise for the rotation it results a higher error from the IMU sensor alone but still much better from the vision.

The next tables show the error analysis that correspond to the next three cases:

- Case 2 : 2 IMUs and 1 Vision Sensor

	Position error		
	mean (m)	**median** (m)	**std** (m)
Vision	2.1480	2.1946	0.5348
$IMU_{(1)}$	3.8347	3.8244	0.7338
$IMU_{(2)}$	4.0967	4.1724	0.7657
Fusion	1.6816	1.7106	0.3963

TABLE 5.3: Case 2: mean, median and standard deviation for position error for sensors and fusion

	Rotation angle error		
	mean (°)	**median** (°)	**std** (°)
Vision	20.1310	18.4702	8.6150
$IMU_{(1)}$	5.8394	5.1448	2.7851
$IMU_{(2)}$	5.9462	5.6175	2.3784
Fusion	9.6299	9.4746	3.8358

TABLE 5.4: Case 2: mean, median and standard deviation for rotation angle error for sensors and fusion

In the second case we may see from the error analysis that the results follow the same logic as the first case. The mean error in position for the fusion is 1.6816 m with standard deviation of 0.3963 m than in the vision alone which is 2.1480 m and 0.5348 m accordingly. Of course the values for the IMUs error are higher as we have more noise in the IMU measurements. Regarding the angle error we have also an improvement in the mean error from 20.1310° with 8.6150° standard deviation, to 9.6299° with 3.8358° standard deviation for the fusion method accordingly. Again the IMUs solely have smaller errors as we described in the first case.

- Case 3 : 1 IMU and 2 Vision Sensors

	Position error		
	mean (m)	**median** (m)	**std** (m)
$\text{Vision}_{(1)}$	2.1480	2.1946	0.5348
$\text{Vision}_{(2)}$	2.2099	2.2410	0.4449
IMU	3.8347	3.8244	0.7338
Fusion	1.5689	1.5939	0.3312

TABLE 5.5: Case 3: mean, median and standard deviation for position error for sensors and fusion

	Rotation angle error		
	mean (°)	**median** (°)	**std** (°)
$\text{Vision}_{(1)}$	20.1310	18.4702	8.6150
$\text{Vision}_{(2)}$	19.3277	18.7899	7.6916
IMU	5.8394	5.1448	2.7851
Fusion	9.5670	9.6979	4.0265

TABLE 5.6: Case 3: mean, median and standard deviation for rotation angle error for sensors and fusion

In third case we have more vision based sensors and the fusion results as we can see from the error values are more precise in position, as the mean error is 1.5689 m with standard deviation of 0.3312 m. One notice is that for the rotation angle error the values of mean error that is 9.5670° with standard

deviation of 4.0265°, are only slightly better than the second case which is natural as the vision sensors are assumed to have worse performance than the IMUs, but still better than the second case.

The error is very close to the second case which implies that the three sensor fusion (two vision and one IMU) information are improving the results for rotation even though the vision sensors give more noisy measurements. Naturally the vision sensors solely have much greater errors, with the first vision sensor to have a mean error of 20.1310° with 8.6150° standard deviation and second vision to have 19.3277° mean and 7.6916° standard deviation.

- Case 4 : 2 IMUs and 2 Vision Sensors

	Position error		
	mean (m)	**median** (m)	**std** (m)
$\text{Vision}_{(1)}$	2.1480	2.1946	0.5348
$\text{Vision}_{(2)}$	2.2099	2.2410	0.4449
$\text{IMU}_{(1)}$	3.8347	3.8244	0.7338
$\text{IMU}_{(2)}$	4.0967	4.1724	0.7657
Fusion	1.5613	1.5895	0.3351

TABLE 5.7: Case 4: mean, median and standard deviation for position error for sensors and fusion

	Rotation angle error		
	mean (°)	**median** (°)	**std** (°)
$\text{Vision}_{(1)}$	20.1310	18.4702	8.6150
$\text{Vision}_{(2)}$	19.3277	18.7899	7.6916
$\text{IMU}_{(1)}$	5.8394	5.1448	2.7851
$\text{IMU}_{(2)}$	5.9462	5.6175	2.3784
Fusion	9.2223	9.4418	3.8930

TABLE 5.8: Case 4: mean, median and standard deviation for rotation angle error for sensors and fusion

In this last case we have all the information from sensors and as expected the mean error of 1.5613 m and standard deviation of 0.3351 m for the position, and

mean error of 9.2223° with standard deviation of 3.8930° for rotation are the lowest than the other cases with two or three fused sensors. The only difference that we observe is that the standard deviation for the position error in the fourth case is 0.3351 m and in the third case is 0.3312 m and this is the only case that we have a lower standard deviation value between the two cases but this is a very small difference which may be from the random noise that we introduced in our synthetic data.

In the below figures we show the mean error comparison between the four cases and how this is changing as we add the different sensors.

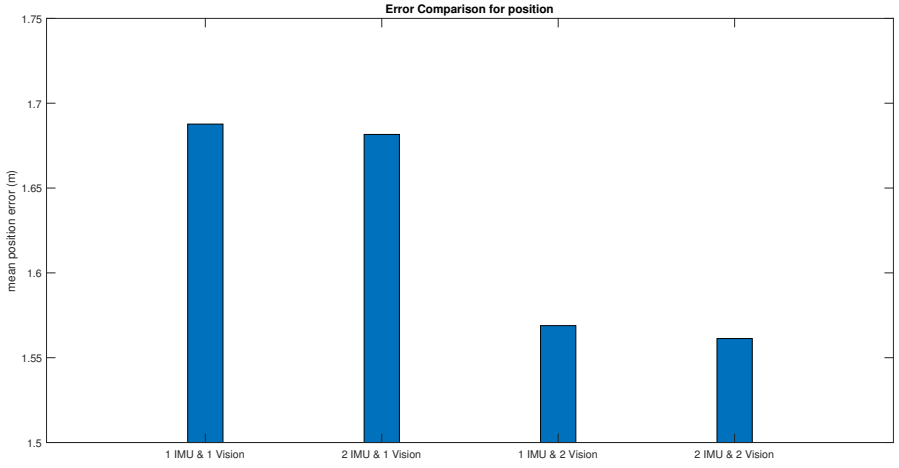


FIGURE 5.7: Error comparison for position

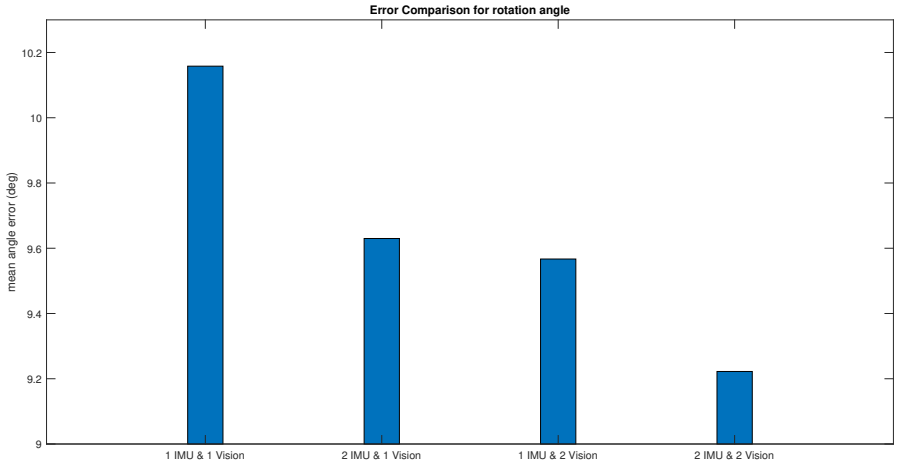


FIGURE 5.8: Error comparison for rotation angle

In the above bar graphs, it is clearly shown that we have a decrease in the mean error in the position from the first to the last case. As we add the second IMU in the second case we see that there is not much of improvement as the mean error drops only from 1.6877 m to 1.6816 m. However when the second vision sensor is inserted the error drops to 1.5689 m and in the last case it takes the minimum value of 1.5613 m.

Regarding the rotation we see from the figure 5.8 that the mean error is immediately decreasing from 10.1581° to 9.6299° and then it is further decreased so in the fourth case with the four sensors it takes the value of 9.2223°.

5.1.2 Results for occlusion case

As we have described in Case 5 : Occlusion, we stop the sequence of the vision-based sensor for 20 poses in a random position of the trajectory. In this time interval the filter is relying only on the IMUs and the process model prediction. In the following figures 5.9 - 5.14 we may see the behaviour of the filter when the occlusion occurs from the 35th to the 55th pose for both the rotation and translation component.

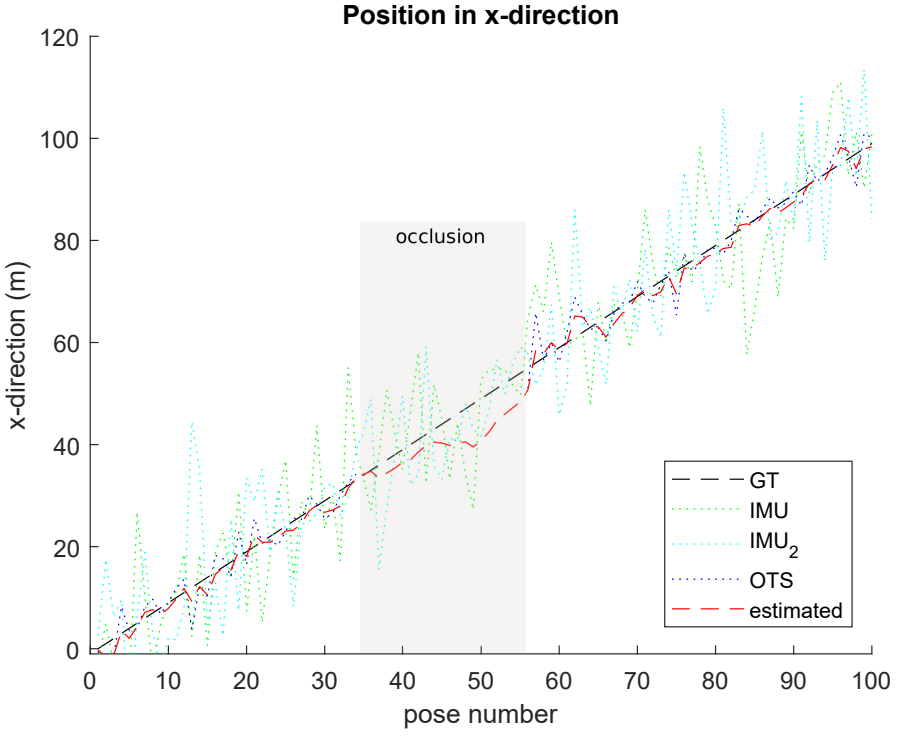


FIGURE 5.9: Occlusion case, position in x-direction

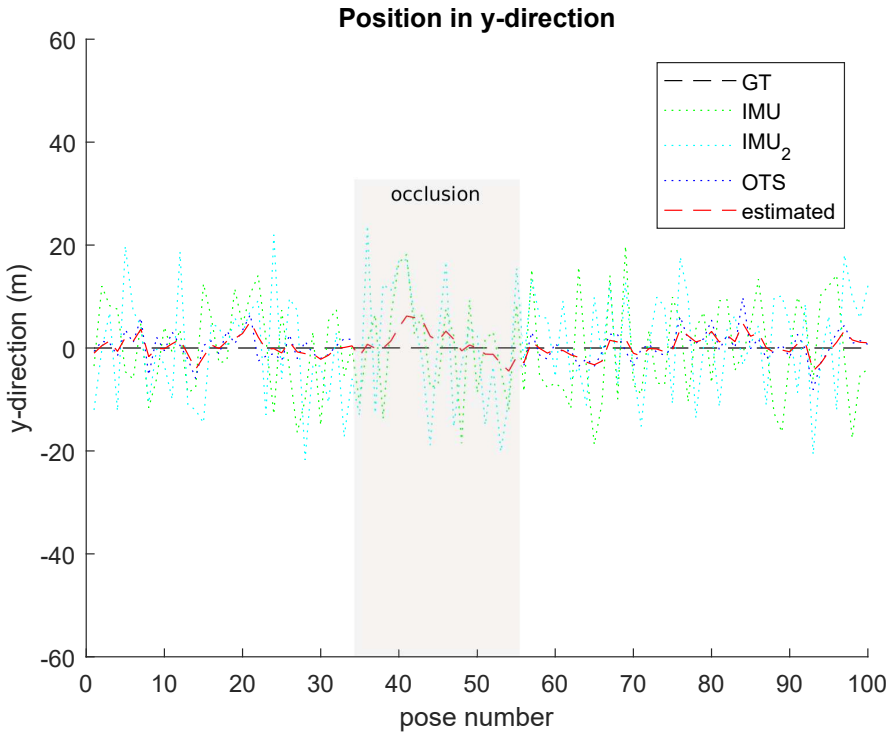


FIGURE 5.10: Occlusion case, position in y-direction

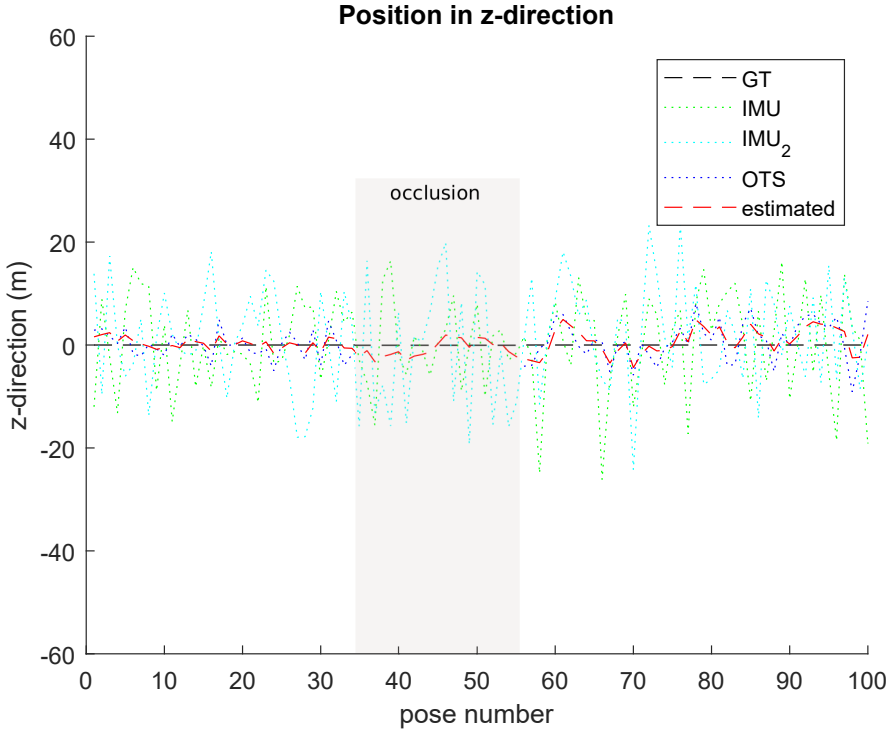


FIGURE 5.11: Occlusion case, position in z-direction

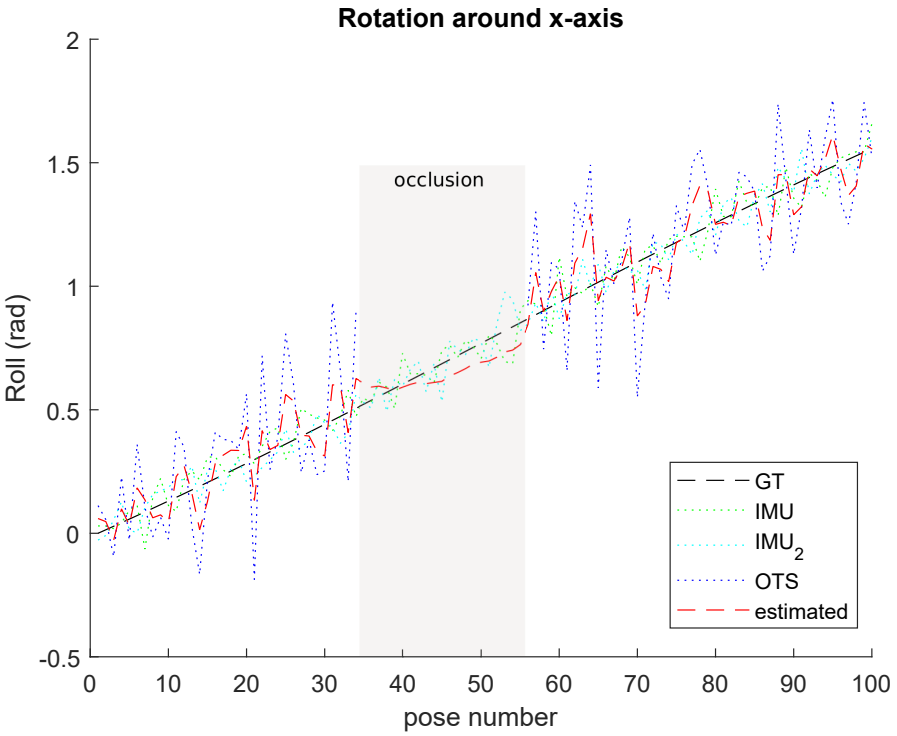


FIGURE 5.12: Occlusion case, rotation around x-axis

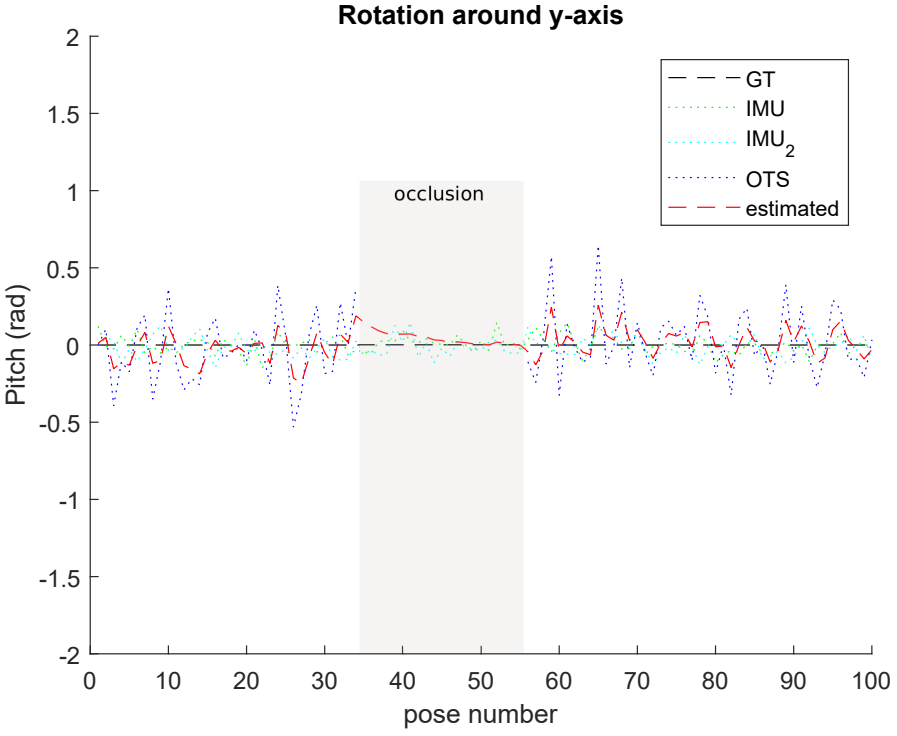


FIGURE 5.13: Occlusion case, rotation around y-axis

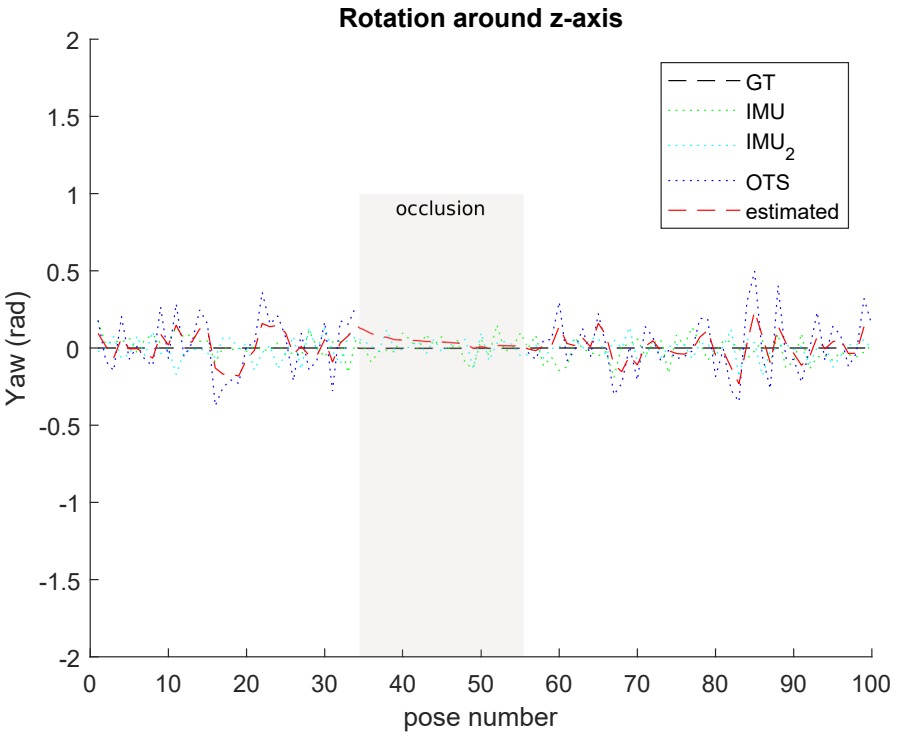


FIGURE 5.14: Occlusion case, rotation around z-axis

In the above figures we can observe that the filter during the loss of line of sight behaves satisfactorily, although especially for the position components the IMUs that we rely on, give much more noisy measurements. This is because the process model is predicting the next pose and with the correction of the data from the IMU it gives a fair expected result. For the rotation we have a better behaviour as it is expected as the rotation variance is lower for the IMUs.

5.2 Dataset results

5.2.1 Results from dataset, no occlusion

In this section we present the results from the experiments that we have made using the whole data from the PennCOSYVIO dataset [9]. Figures 5.15 - 5.17 show the IMU, vision and fusion trajectories along with the ground truth regarding the position and figures 5.18 - 5.20 show the according rotation components in x, y and z axis.

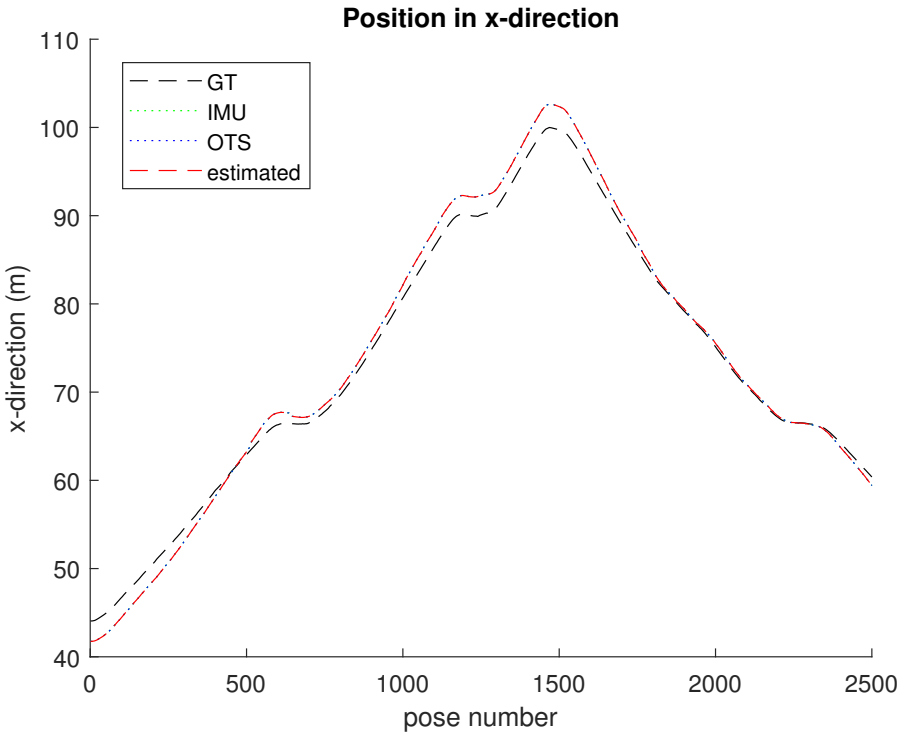


FIGURE 5.15: Dataset, position in x-direction

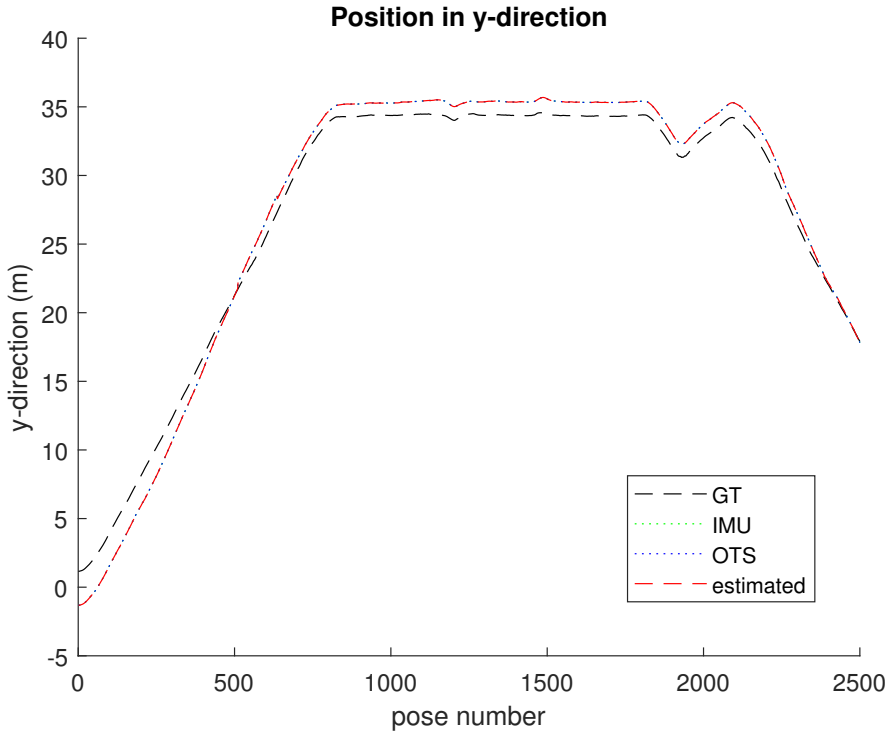


FIGURE 5.16: Dataset, position in y-direction

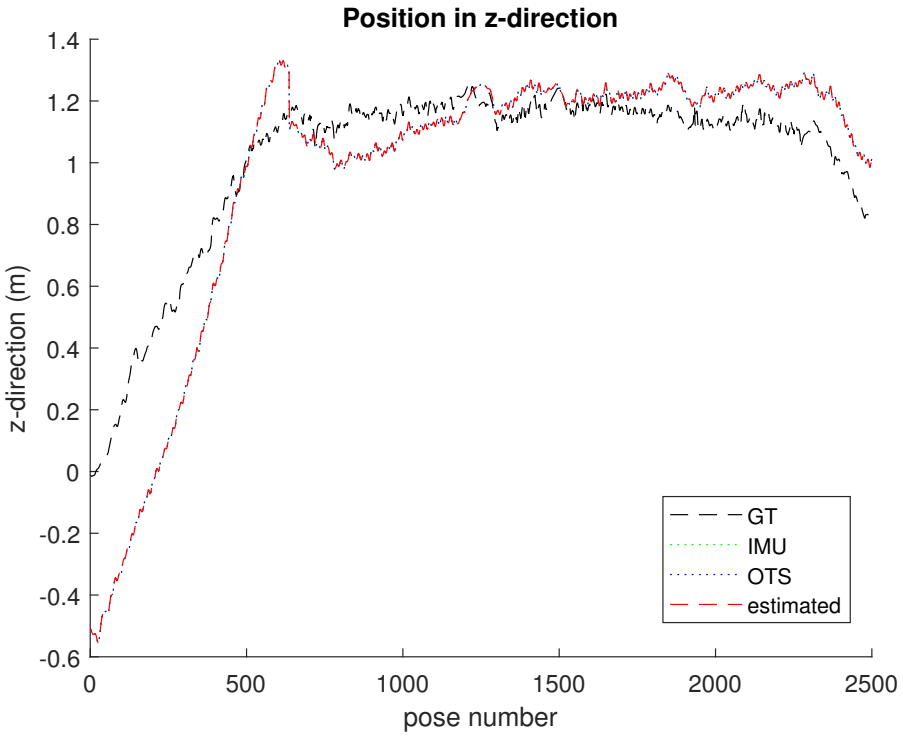


FIGURE 5.17: Dataset, position in z-direction

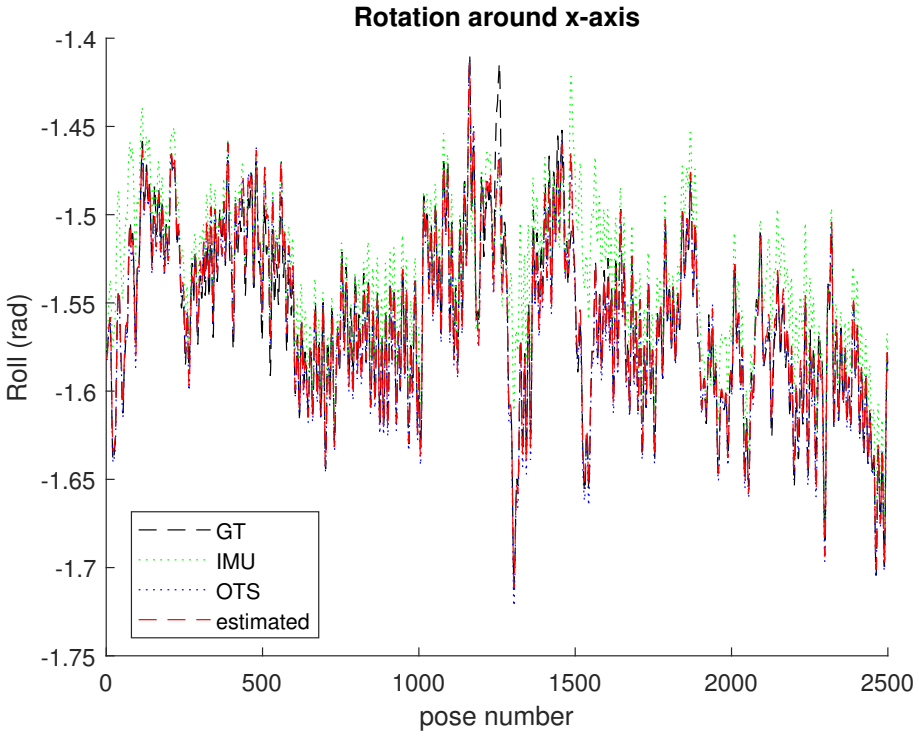


FIGURE 5.18: Dataset, rotation around x-axis

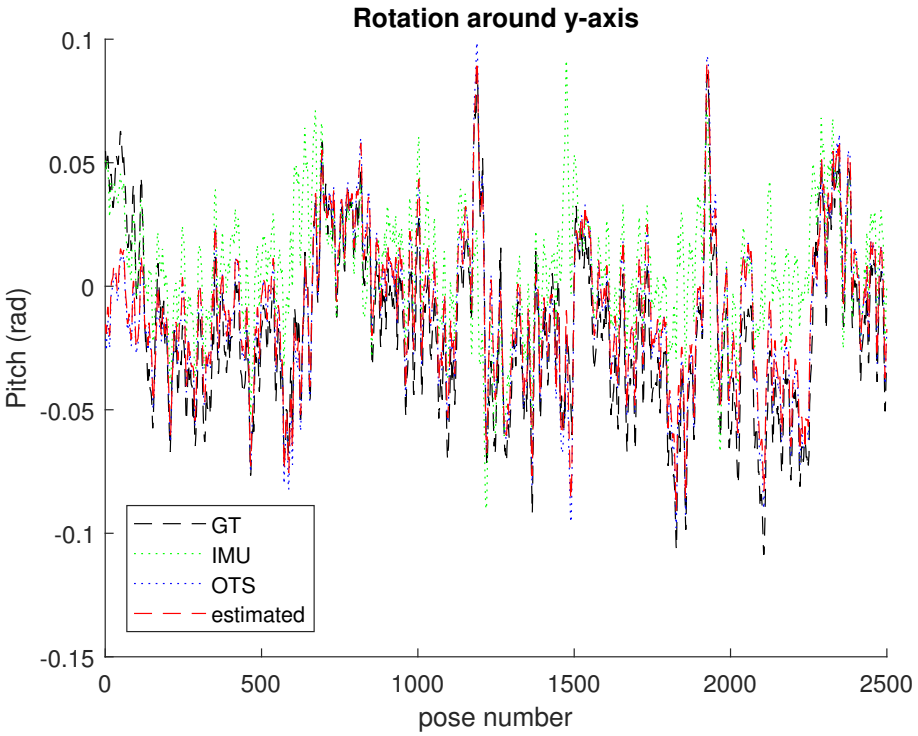


FIGURE 5.19: Dataset, rotation around y-axis

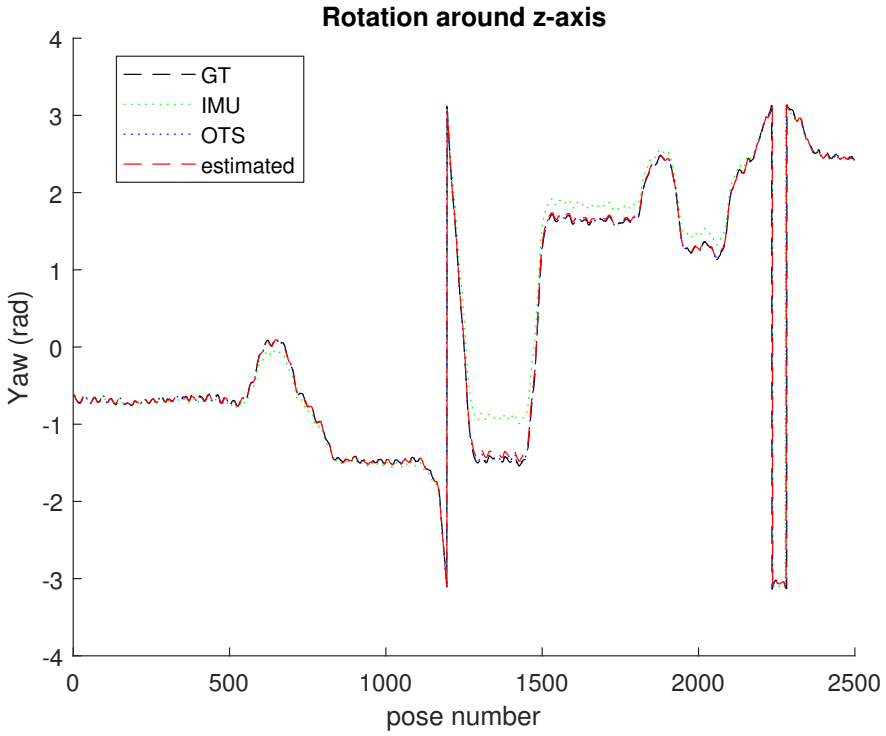


FIGURE 5.20: Dataset, rotation around z-axis

The first notice we should make is about the IMU sensor. As we want the filter to be as effective as possible, for this case that we have a good trajectory from the vision sensor we use it also as an input for the IMU translation component and this is why the green line of IMU is not appearing in the graph. It is just identical with the vision sensor trajectory. As we have explained in section 1.3.2, IMU sensors are not ideal to estimate the position of a tracked object because they drift very fast and that is why we make the above conciliation.

The ground truth values provided by the authors were acquired by markers that were spread in the whole route, inside and outside the building may be not very reliable for benchmarking.

One thing to mention is in figures 5.18 and 5.19, where we can see that there is not much rotation around x and y axis in contrary to figure 5.20 where the most action takes place.

Also there are some pose intervals in figure 5.20 like from 600th-700th pose, from 1300th-1500th pose, or from 1500th-1800th pose that the rotation is rather stable and we can see that the IMU measurements have a small offset regarding the vision and ground truth values.

The following tables show the quantitative error analysis for the whole dataset (no occlusion), presenting the corresponding mean, median and standard deviation of the error of each sensor and the fused method for position and rotation:

	Position error		
	mean (m)	**median** (m)	**std** (m)
Vision	1.2581	1.1930	0.3173
IMU	1.2581	1.1930	0.3173
Fusion	1.2583	1.1929	0.3174

TABLE 5.9: Dataset (no occlusion): mean, median and standard deviation for position error for sensors and fusion

	Rotation angle error		
	mean (°)	**median** (°)	**std** (°)
Vision	1.1817	1.0721	0.6321
IMU	8.0291	4.1650	8.2898
Fusion	1.5397	1.2741	0.9784

TABLE 5.10: Dataset (no occlusion): mean, median and standard deviation for rotation angle error for sensors and fusion

It is obvious from the above tables that the fusion method is not improving the position and this is rather expected as the translation for the IMU is the same with the vision sensor and so there is a very small difference in fusion error. Other than that the values are almost identical.

For the rotation angle error we may see that as the IMU measurements are worse than those from the vision based sensor the fusion method is way better than those from the IMU with a mean error of 1.5397° and standard deviation of 0.9784° comparing with the corresponding mean of 8.0291° with standard deviation of 8.2896° of the IMU, but higher that the vision error values of 1.1817° mean and 0.6321 standard deviation. These results show that when the rotation measurements from vision are better than the OTS we cannot improve the overall error.

5.2.2 Results from dataset, occlusion case

The results from the occlusion sub-case regarding the dataset are presented in the next figures. We are showing specific regions of the whole graph with a grey box when the occlusion occurs as we want to focus on the behaviour of the filter at these poses.

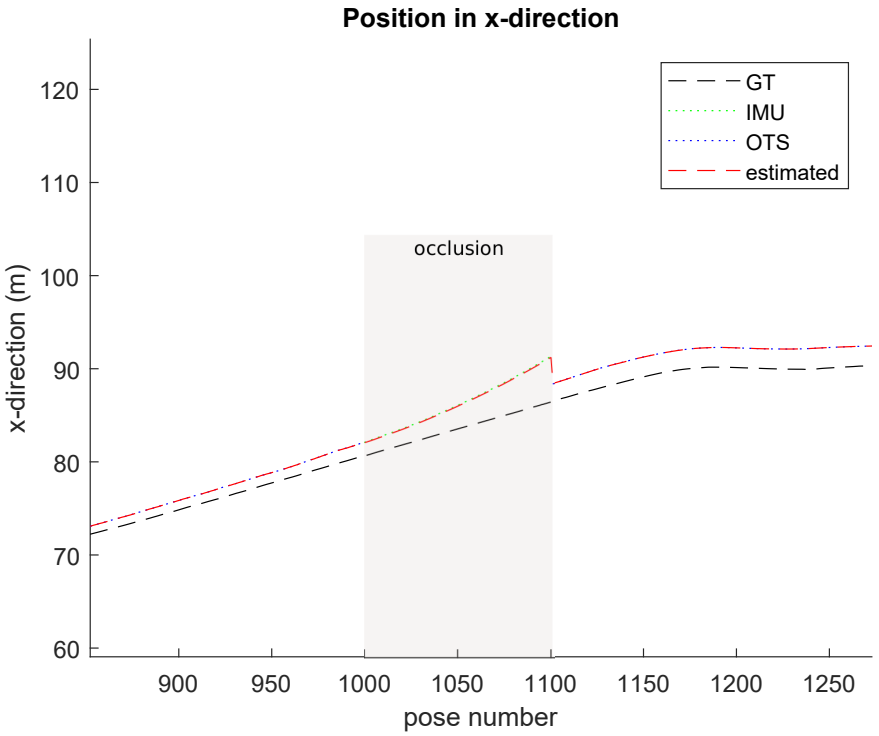


FIGURE 5.21: Dataset occlusion, position in x-direction

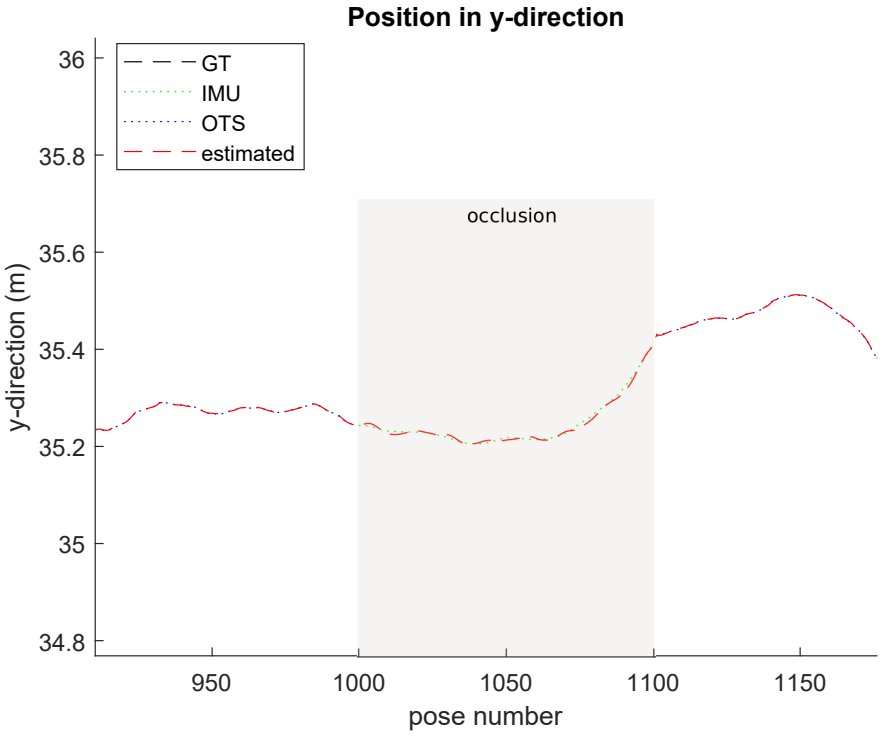


FIGURE 5.22: Dataset occlusion, position in y-direction

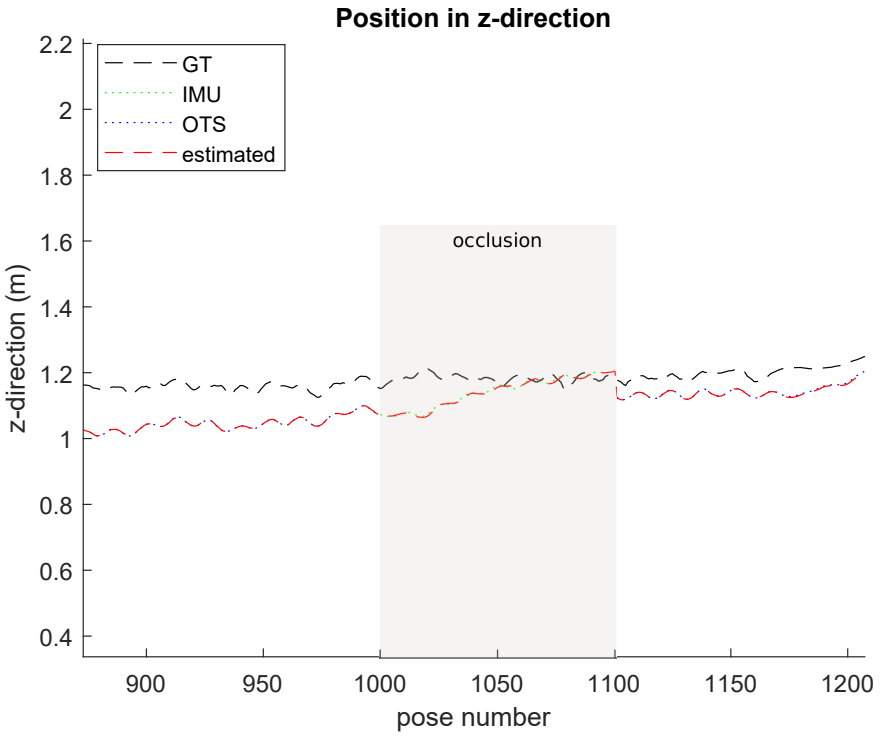


FIGURE 5.23: Dataset occlusion, position in z-direction

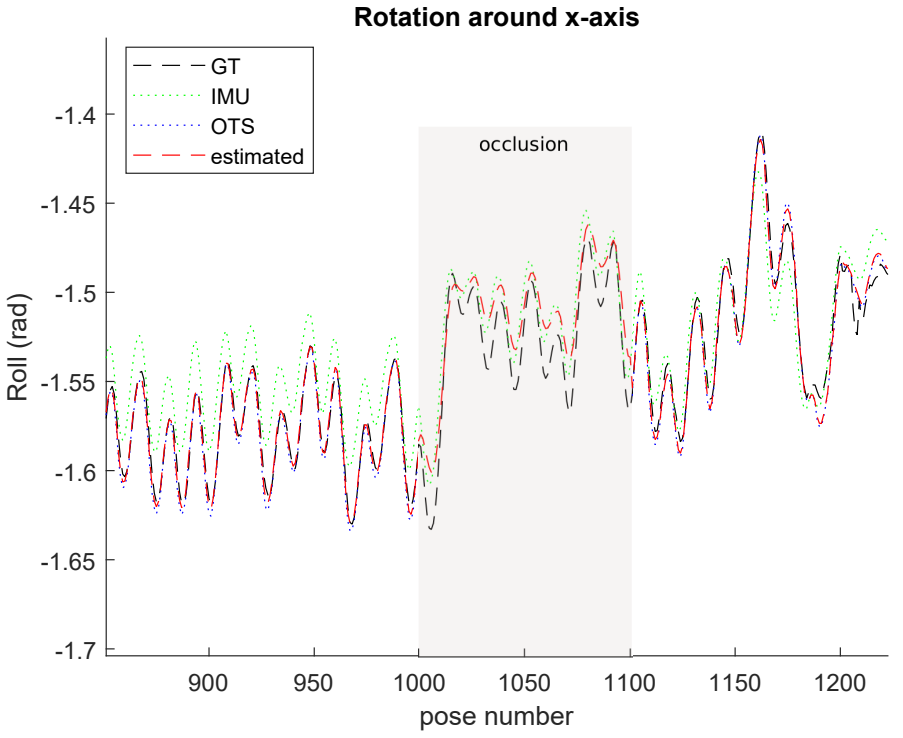


FIGURE 5.24: Dataset occlusion, rotation around x-axis

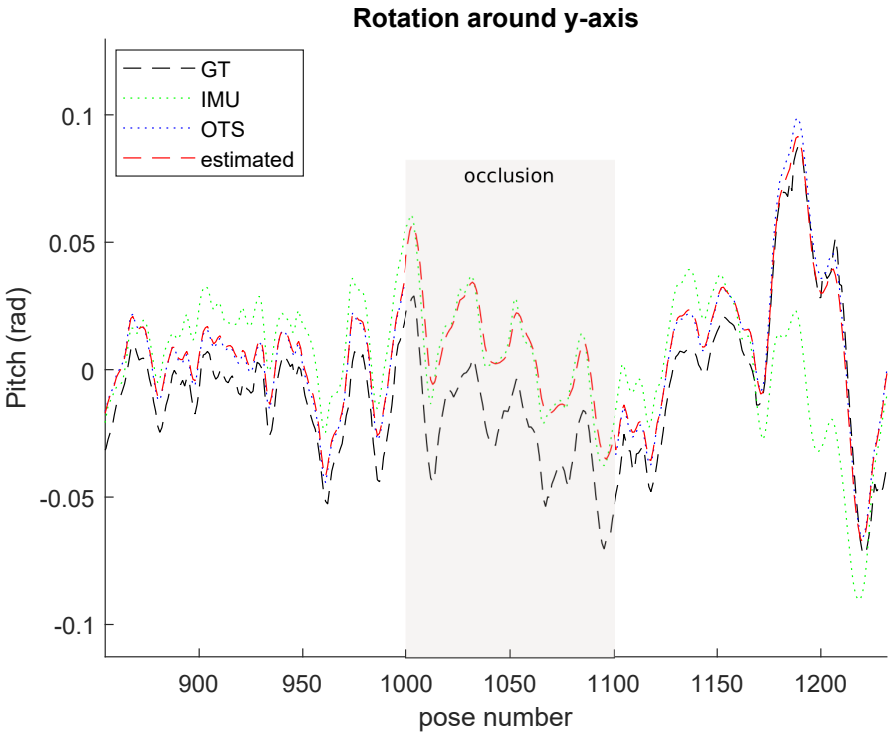


FIGURE 5.25: Dataset occlusion, rotation around y-axis

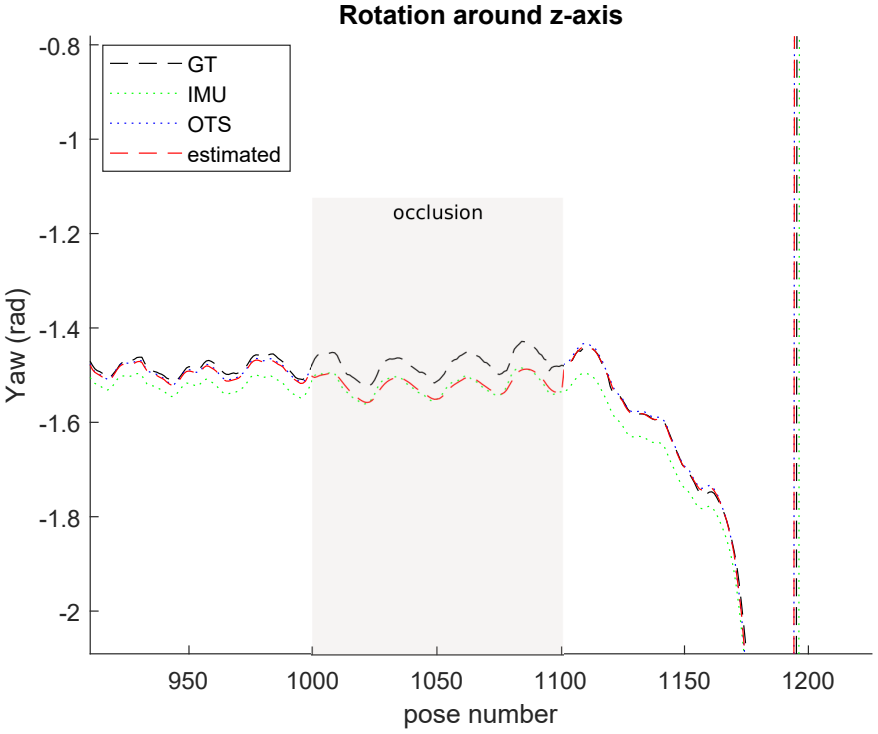


FIGURE 5.26: Dataset occlusion, rotation around z-axis

Form the above figures we make state for the position that during occlusion in x-direction for the first 50 poses the estimated trajectory is following the same direction as the vision and afterwards is staring drifting and increasing rapidly until it reaches the 100 occluded pose pose and then when we return to the normal case, it is the same as the vision again. The same behaviour is also in the z-direction. On the other hand for y-direction during the occlusion the fusion method seems to underestimate the position but it is still stable with not much variation.

For the rotation around x and y axis the figures show better correspondence with the ground truth as the calculation of the rotation is more accurate. As for the rotation around z-axis the estimated values are close to the IMU measurements and as these are calculated with more precision, the line is following the same shape but with a small offset.

In the following figures we show the error against the number of the poses for the position and the rotation.

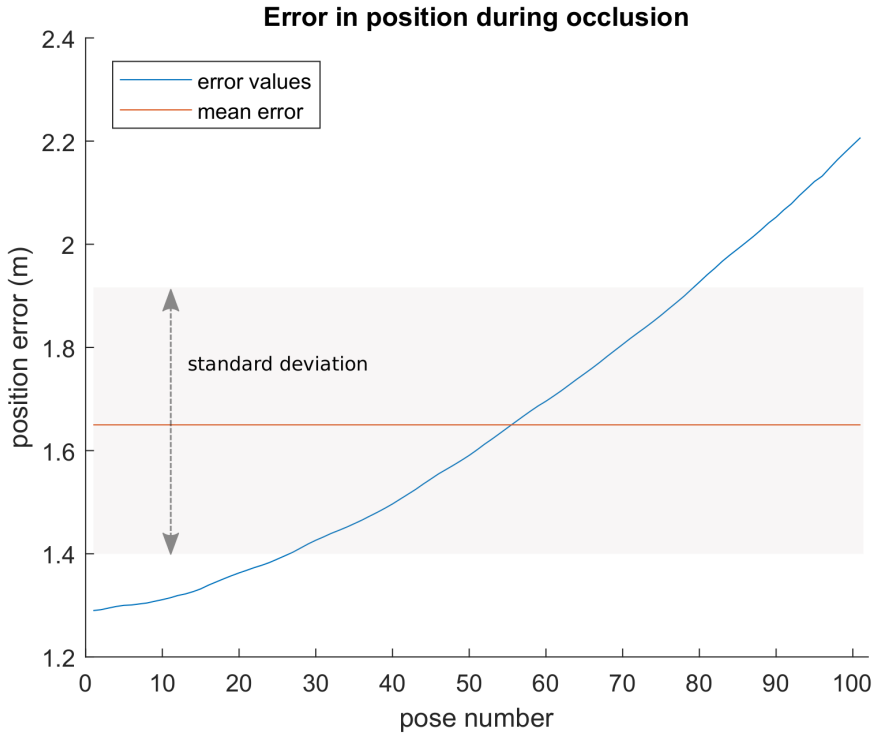


FIGURE 5.27: Dataset occlusion, error in position

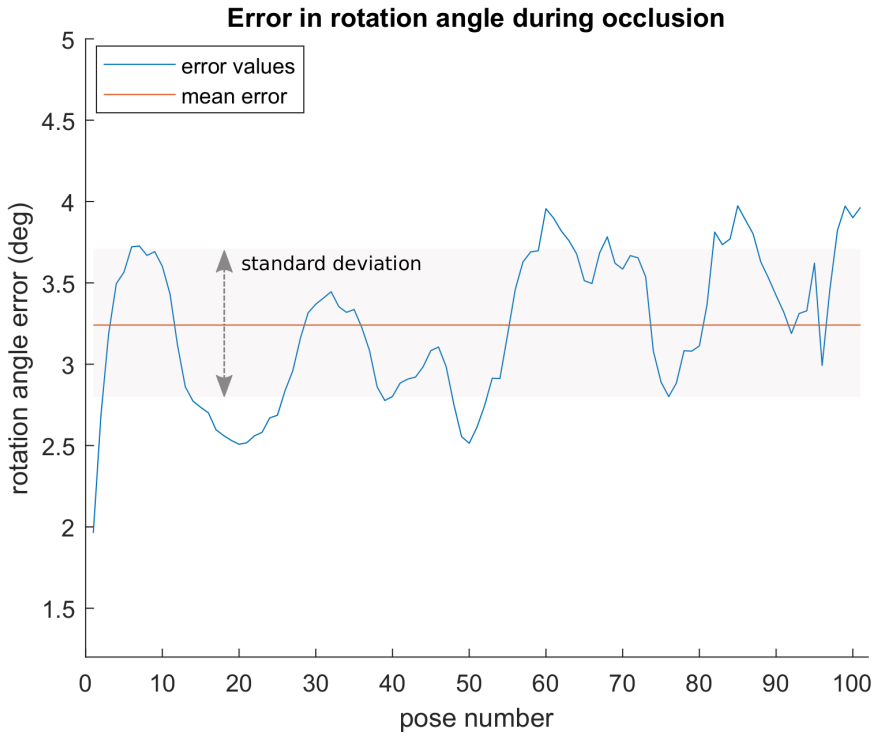


FIGURE 5.28: Dataset occlusion, error in rotation angle

From the first figure it is obvious that the error in position is increasing exponentially, something that is in accordance with the statements that we already did about the behaviour of the IMU regarding the position in section 1.3.2. The measurements of the IMU are drifting as the time passes and at last pose of the occlusion the error is already 2.2 m compared to the mean error of 1.2583 with standard deviation of 0.3174 that we see in the non occlusion case in table 5.9 and it is continuously increasing. The time of the occlusion is 3s and it is obvious that if the occlusion would last longer the error would be significant higher.

On the other hand we may see that the error in the angle is increasing much slower, with an oscillated behaviour and this is because the readings from the IMU are way more reliable for the rotation than the position. The error at the end of the occlusion for the rotation angle is 3.9643° which shows that the occlusion still increases the error in comparison with the mean error of 1.5397° and standard deviation of 0.9784 (table 5.10), but we can have a fair estimation of the rotation in this case that the IMU measurements for the rotation are worse than those from the vision sensor as the error is around a mean value of 3.2407°.

5.3 Real data results

In the next three figures we present the rotation around x, y and z axes for the real data from vision sensor, IMU and fusion method.

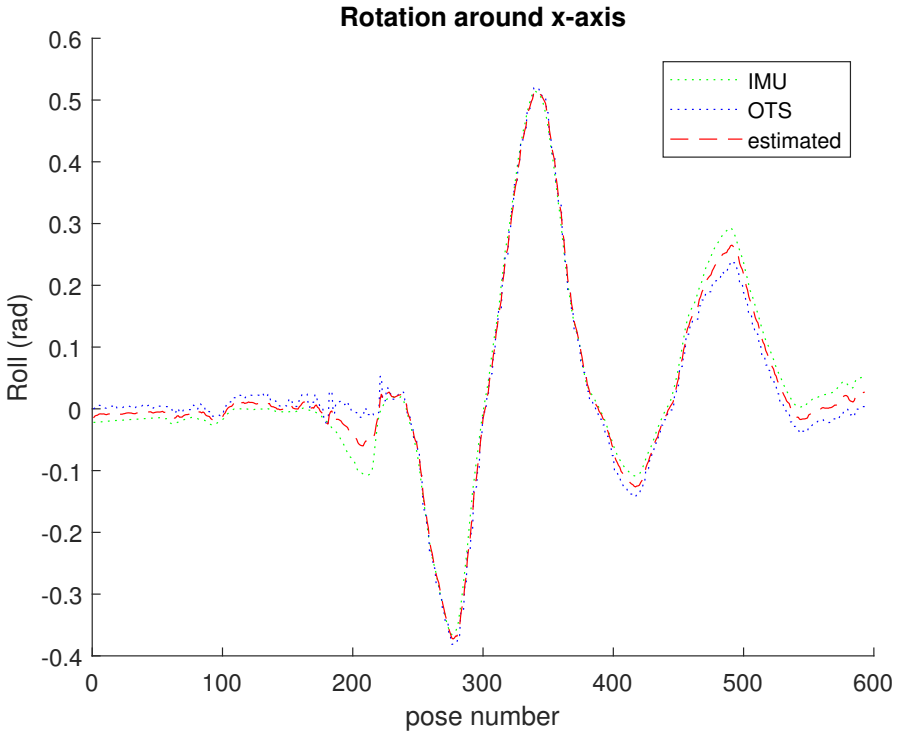


FIGURE 5.29: Real Data, rotation around x-axis

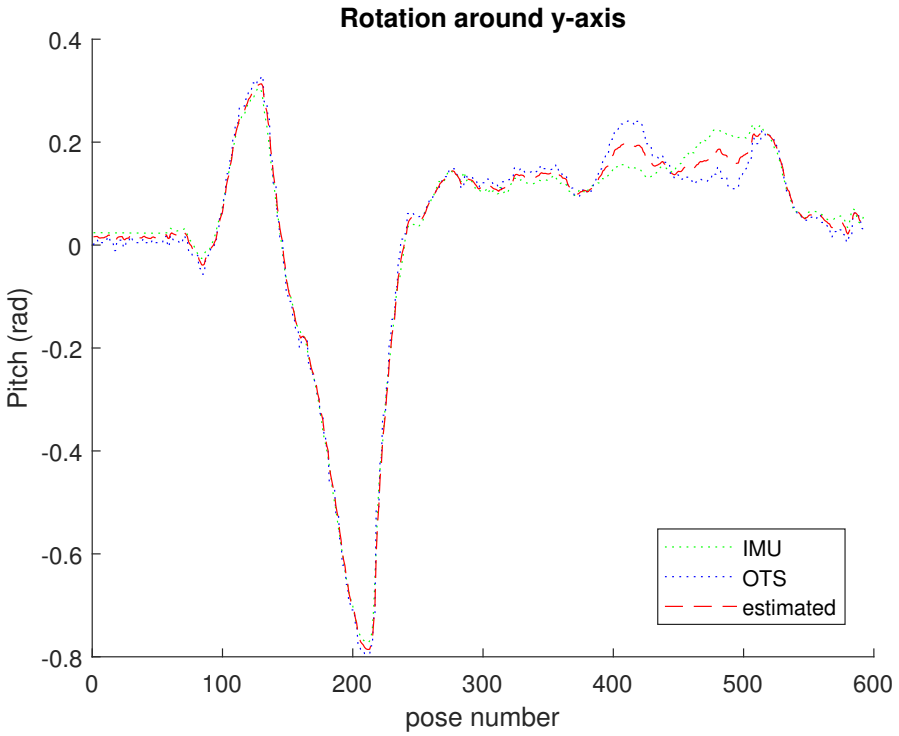


FIGURE 5.30: Real Data, rotation around y-axis

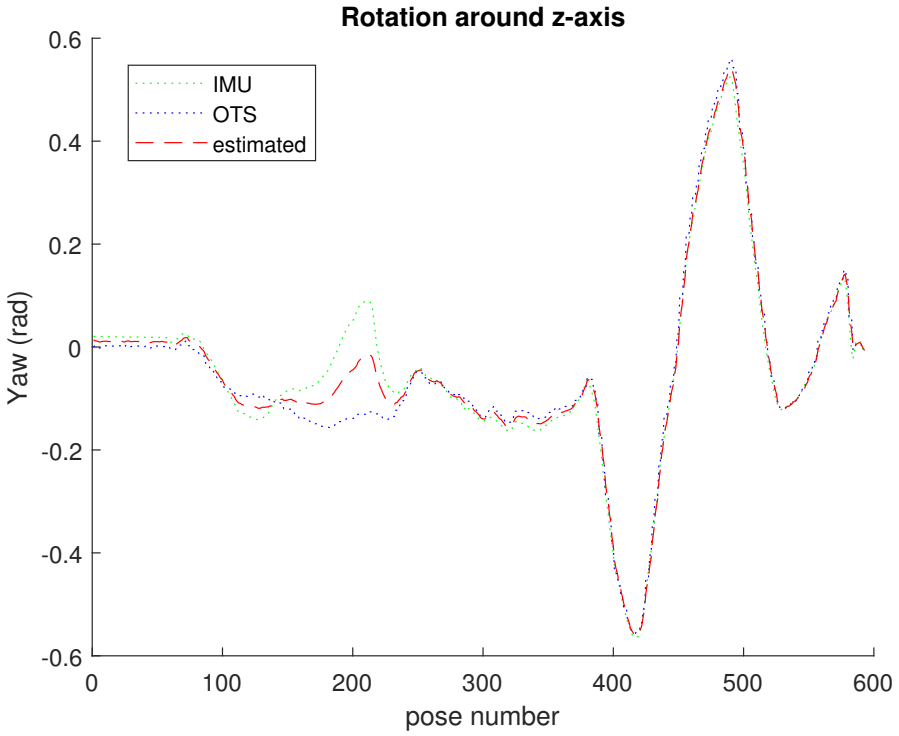


FIGURE 5.31: Real Data, rotation around z-axis

The above figures show that the filter is able to fuse the data from the IMU and the vision based sensor and the estimated values are smoother that those of the vision sensor itself, giving a mean estimation of the values of the two sensors.

Chapter 6

Conclusion and Discussion

In this scientific study we implement an EKF filter in order to fuse the data from different sensors, inertial and vision-based. We use the dual quaternion formalization which gives us not only the advantage of coupling the translation and the rotation of the tracked object(s), but also we avoid kinematic singularities which may occur by using the common Euler angles for the representation and calculations of the rotation. The implementation is made for up to four sensors fusion and we also provide a method for n sensors fusion. For the evaluation of our method we test the filter with synthetic and real data and also we include the case that we loose information from the vision-based sensor.

In the following sections we emphasize to the most interesting and valuable results. From these results, conclusions are drawn and suggestions for possible future research are proposed.

6.1 Discussion of the results

Regarding the synthetic data we find out that the for the position as we add more sensors the mean error for the trajectory is decreasing as it shown in figure 5.7. The estimation is not changing drastically from 1st to 2nd case as the added sensor is an IMU which has higher error in its measurements, but when a vision based sensor is added to the system, the mean error is decreasing from 1.6877 m in the first case to 1.5613 m in the last case with the four sensors. About the rotation, we may see from the comparison figure 5.8 that the filter is increasing

the accuracy by reducing the mean error from 10.1581° in the first case with two sensors to 9.2223° for the last case with four sensors fusion. The angle error decreases according to what kind of sensor that is added. More accurate is the output when we have two IMUs and this is natural as we consider them less noisy for the rotation measurements than the vision based ones.

The results from occlusion case for the synthetic data show that the filter estimates fairly good the position with the error in the beginning to be 2.1319 m and at the end 3.2862 m. For the rotation angle the error at the end is 5.1375°, about half of that from the beginning which is 11.9112°. This is an expected result as during the occlusion we rely only on the IMUs measurements with a much better accuracy from that of the vision based sensor.

The results of the experiments with the data from the dataset that we use it is obvious from table 5.9 that as the translation components are considered the same for vision and IMU we have no change in the position error. As the data from the vision based sensor for the rotation are more accurate than the ones from the IMU, the fusion method gives slightly bigger mean error for the rotation, from 1.1817° of the vision to 1.5397°.

The filter during the occlusion in the dataset case, for the position is giving an exponential increasing error as shown in figure 5.27, which starts from 1.2899 m up to 2.2068 m with an increasing trend. This is a natural result because of the drift of the IMU measurements for the translation components. For the rotation angle we observe also an increase in the error from 1.9627° in the beginning of the occlusion to 3.9643° at the end of it, but with the major difference that the error values are not increasing exponentially but we have an oscillated behaviour as it is shown in figure 5.28. That shows that the measurements from the IMUs help, as they are more reliable for the estimation of the rotation, to have better accuracy for the orientation during the occlusion.

Regarding the real data we see from the figures 5.29 - 5.31 that the filter is smoothing the data from IMU and vision sensors. As we don't count on the translation components from the IMU, as they are extremely unreliable, we visualize only the rotation around each axis.

At this point we should mention that the choice of the parameters for the filter plays a huge role for the output results. We must adjust the covariance values for every sensor according to how noisy the sensor is or how stable it is. The

parameters for all the above experiments were found with numerous empirical tests that we made in order to find the appropriate combinations for the filter to give the most efficient estimations, which leaves room for improvement to adjust them online.

6.2 Conclusion

As a conclusion we can say that our method of multiple sensor fusion appear to be effective in the estimation of position and rotation for the synthetic data, as the mean error in position is decreasing from 2.1480 m when we have only the vision sensor to 1.5613 m when we have the data fusion of four sensors, two vision and two IMUs. When it comes to the real data we see that when the data from the vision sensor are very precise the filter is not increasing the accuracy of the system. On the other hand for the occlusion case, for short periods of time of about 3 seconds the filter is giving good results both for orientation and position as long as the data from the inertial sensor for translation are used only when we have loss of line of sight.

Finally we should take into account that there should be a compromise between translation and rotation estimation. Whenever we have a sensor that has a better precision for rotation and worse for translation like an IMU, we should consider that if we rely more in that sensor we gain accuracy in orientation but have lower accuracy in position.

6.3 Future work

Because of the limitations that we have already mentioned, there is a lot of space for improvement of the current filter and also for expanding the potentials of the multiple fusion method. Firstly there should be a method to evaluate each sensor that is inserted in the system and so the error parameters could adjust accordingly. The estimation for the position from the inertial sensors could be improved in order to increase the accuracy of the system both in the occlusion case, where we can prolong the time of the good estimation of the poses during the lost of line of sight and for the no occlusion cases. Also the method for the frequency augmentation of

the vision sensor(s) may be improved to give more accurate results. The method should be also evaluated by different types of sensors and methods for estimating the position and orientation. Finally the filter may be used also in real time and be compared with other methods that use different filters and mathematical approaches.

Bibliography

[1] Kriti Kumar, Ashley Varghese, Pavan K Reddy, N Narendra, Prashanth Swamy, M Girish Chandra, and P Balamuralidhar. An improved tracking using imu and vision fusion for mobile augmented reality applications. *arXiv preprint arXiv:1411.2335*, 2014.

[2] Changyu He, Peter Kazanzides, Hasan Tutkun Sen, Sungmin Kim, and Yue Liu. An inertial and optical sensor fusion approach for six degree-of-freedom pose estimation. *Sensors*, 15(7):16448–16465, 2015.

[3] Kun Zhang, Zaojun Fang, Jianran Liu, Zhengxing Wu, and Min Tan. Fusion of vision and imu to track the racket trajectory in real time. In *Mechatronics and Automation (ICMA), 2017 IEEE International Conference on*, pages 1769–1774. IEEE, 2017.

[4] Hyunmin Oh, You Seong Chae, Jinung An, and Min Young Kim. Pose estimation of surgical instrument using sensor data fusion with optical tracker and imu based on kalman filter. In *MATEC Web of Conferences*, volume 32. EDP Sciences, 2015.

[5] https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTAMlRhFpPlAM-4MahgoC8ZsvzfDuAkEodZy5SHBKJESaZOkJq9tA. Polaris camera with passive marker tool, accessed:17-April-2018.

[6] https://upload.wikimedia.org/wikipedia/commons/1/15/ETHOS_pcb.png. Inertial Measurement Unit with all its components, accessed:17-April-2018.

[7] https://wigi725ylkk16hjts2tbfo9i-wpengine.netdna-ssl.com/medical/wp-content/uploads/sites/4/2013/09/AuroraFGs.jpg. Aurora AC-based Electromagnetic tracking system, accessed:17-April-2018.

[8] Greg Welch and Gary Bishop. An introduction to the kalman filter. department of computer science, university of north carolina. *ed: Chapel Hill, NC, unpublished manuscript*, 2006.

[9] Bernd Pfrommer, Nitin Sanket, Kostas Daniilidis, and Jonas Cleveland. Penncosyvio: A challenging visual inertial odometry benchmark. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 3847–3854, 2017. doi: 10.1109/ICRA.2017.7989443. URL https://doi.org/10.1109/ICRA.2017.7989443.

[10] Tobias Sielhorst, Marco Feuerstein, and Nassir Navab. Advanced medical displays: A literature review of augmented reality. 4:451 – 467, 01 2009.

[11] Andrew D Wiles, David G Thompson, and Donald D Frantz. Accuracy assessment and interpretation for optical tracking systems. In *Medical Imaging 2004: Visualization, Image-Guided Procedures, and Display*, volume 5367, pages 421–433. International Society for Optics and Photonics, 2004.

[12] Rasool Khadem, Clement C Yeh, Mohammad Sadeghi-Tehrani, Michael R Bax, Jeremy A Johnson, Jacqueline Nerney Welch, Eric P Wilkinson, and Ramin Shahidi. Comparative tracking error analysis of five different optical tracking systems. *Computer Aided Surgery*, 5(2):98–107, 2000.

[13] Patrick Haggard and Alan M Wing. Assessing and reporting the accuracy of position measurements made with optical tracking systems. *Journal of Motor Behavior*, 22(2):315–321, 1990.

[14] X. Yun, E. R. Bachmann, and R. B. McGhee. A simplified quaternion-based algorithm for orientation estimation from earth gravity and magnetic field measurements. *IEEE Transactions on Instrumentation and Measurement*, 57 (3):638–650, March 2008. ISSN 0018-9456. doi: 10.1109/TIM.2007.911646.

[15] Wilson E. Cleary K. Birkfellner W., Hummel J. Tracking devices. In *Image-Guided Interventions*, chapter 2, pages 23–44. Springer, Boston, MA, 2008.

[16] João Paulo Lima, Rafael Roberto, Francisco Simões, Mozart Almeida, Lucas Figueiredo, João Marcelo Teixeira, and Veronica Teichrieb. Markerless tracking system for augmented reality in the automotive industry. *Expert Systems with Applications*, 82:100–114, 2017.

[17] Oliver J Woodman. An introduction to inertial navigation. Technical report, University of Cambridge, Computer Laboratory, 2007.

[18] Isaac Skog and Peter Händel. Calibration of a mems inertial measurement unit. In *XVII IMEKO World Congress*, pages 1–6, 2006.

[19] Stephan Weiss, Markus W Achtelik, Simon Lynen, Michael C Achtelik, Laurent Kneip, Margarita Chli, and Roland Siegwart. Monocular vision for long-term micro aerial vehicle state estimation: A compendium. *Journal of Field Robotics*, 30(5):803–831, 2013.

[20] Peiyi Y. Chunyang Z. Yuyang, W. and L. Zheming. A dual quaternion based fusion framework for imu data with 6 dof pose. *Journal of Electronics and Information Science*, 1(1):22–31, 2016.

[21] Neda Parnian, Seong-hoon Peter Won, and Farid Golnaraghi. Position sensing using integration of a vision system and inertial sensors. In *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE*, pages 3011–3015. IEEE, 2008.

[22] Jeroen D Hol, Thomas B Schön, Henk Luinge, Per J Slycke, and Fredrik Gustafsson. Robust real-time tracking by fusing measurements from inertial and vision sensors. *Journal of Real-Time Image Processing*, 2(2-3):149–160, 2007.

[23] Yaqin Tao, Huosheng Hu, and Huiyu Zhou. Integration of vision and inertial sensors for home-based rehabilitation. inervis 2005. In *2nd Workshop on Integration of Vision and Inertial Sensors, IEEE Int. Conf. on Robotics and Automation*, 2005.

[24] Leopoldo Armesto, Josep Tornero, and Markus Vincze. Fast ego-motion estimation with multi-rate fusion of inertial and vision. *The International Journal of Robotics Research*, 26(6):577–589, 2007.

[25] Suya You and Ulrich Neumann. Fusion of vision and gyro tracking for robust augmented reality registration. In *Virtual Reality, 2001. Proceedings. IEEE*, pages 71–78. IEEE, 2001.

[26] Joakim Hugmark. Inertial-aided ekf-based structure from motionfor robust real-time augmented reality, 2013.

[27] Nima Enayati, Elena De Momi, and Giancarlo Ferrigno. A quaternion-based unscented kalman filter for robust optical/inertial motion tracking in computer-assisted surgery. *IEEE Transactions on Instrumentation and Measurement*, 64(8):2291–2301, 2015.

[28] Simon J. Julier and Jeffrey K. Uhlmann. A new extension of the kalman filter to nonlinear systems. 3068, 02 1999.

[29] Mathieu St-Pierre and Denis Gingras. Comparison between the unscented kalman filter and the extended kalman filter for the position estimation module of an integrated navigation information system. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 831–835. IEEE, 2004.

[30] Mirko Daniele Comparetti, Elena De Momi, Tim Beyl, Mirko Kunze, Jrg Raczkowsky, and Giancarlo Ferrigno. Convergence analysis of an iterative targeting method for keyhole robotic surgery. *International Journal of Advanced Robotic Systems*, 11(4):60, 2014. doi: 10.5772/58250. URL https://doi.org/10.5772/58250.

[31] Bastian Hartmann, Norbert Link, and Gert F Trommer. Indoor 3d position estimation using low-cost inertial sensors and marker-based video-tracking. In *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, pages 319–326. IEEE, 2010.

[32] Benjamin Busam, Tolga Birdal, and Nassir Navab. Camera pose filtering with local regression geodesics on the riemannian manifold of dual quaternions. *arXiv preprint arXiv:1704.07072*, 2017.

[33] Andreas Tobergte, Mihai Pomarlan, and Gerd Hirzinger. Robust multi sensor pose estimation for medical applications. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 492–497. IEEE, 2009.

[34] Stamatia Giannarou, Zhiqiang Zhang, and Guang-Zhong Yang. Deformable structure from motion by fusing visual and inertial measurement data. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4816–4821. IEEE, 2012.

[35] Sebastian Madgwick. An efficient orientation filter for inertial and inertial/-magnetic sensor arrays. *Report x-io and University of Bristol (UK)*, 25, 2010.

[36] Mingyang Li and Anastasios I Mourikis. High-precision, consistent ekf-based visual-inertial odometry. *The International Journal of Robotics Research*, 32 (6):690–711, 2013.

[37] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[38] Ashley Varghese, M Girish Chandra, and Kriti Kumar. Dual quaternion based imu and vision fusion framework for mobile augmented reality. In *Intelligent Signal Processing (WISP), 2015 IEEE 9th International Symposium on*, pages 1–6. IEEE, 2015.

[39] Manuel Schwaab, Michailas Romanovas, Davide Plaia, Tobias Schwarze, and Yiannos Manoli. Fusion of visual odometry and inertial sensors using dual quaternions and stochastic cloning. In *Information Fusion (FUSION), 2016 19th International Conference on*, pages 573–580. IEEE, 2016.

[40] Nuno Filipe, Michail Kontitsis, and Panagiotis Tsiotras. Extended kalman filter for spacecraft pose estimation using dual quaternions. *Journal of Guidance, Control, and Dynamics*, 38(9):1625–1641, 2015.

[41] Robert C Leishman and Timothy W McLain. Multiplicative extended kalman filter for relative rotorcraft navigation. *Journal of Aerospace Information Systems*, 12(12):728–744, 2014.

[42] James Samuel Goddard and Mongi A Abidi. Pose and motion estimation using dual quaternion-based extended kalman filtering. In *Three-Dimensional Image Capture and Applications*, volume 3313, pages 189–201. International Society for Optics and Photonics, 1998.

[43] Rangaprasad Arun Srivatsan, Gillian T Rosen, D Feroze Naina Mohamed, and Howie Choset. Estimating se (3) elements using a dual quaternion based linear kalman filter. In *Robotics: Science and Systems*, 2016.

[44] Edward Pervin and Jon A Webb. Quaternions in computer vision and robotics. 1982.

[45] Benjamin Busam, Marco Esposito, Benjamin Frisch, and Nassir Navab. Quaternionic upsampling: Hyperspherical techniques for 6 dof pose tracking. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 629–638. IEEE, 2016.

[46] Erik B Dam, Martin Koch, and Martin Lillholm. *Quaternions, interpolation and animation*, volume 2. Datalogisk Institut, Københavns Universitet Copenhagen, 1998.

[47] Fuzhen Zhang. Quaternions and matrices of quaternions. *Linear algebra and its applications*, 251:21–57, 1997.

[48] GR Veldkamp. On the use of dual numbers, vectors and matrices in instantaneous, spatial kinematics. *Mechanism and Machine Theory*, 11(2):141–156, 1976.

[49] Felix Mora-Camino and Carlos Alberto Nunes Cosenza. Fuzzy dual numbers. In *Fuzzy Dual Numbers*, pages 11–16. Springer, 2018.

[50] WB Vasantha Kandasamy and Florentin Smarandache. Dual numbers, 2012.

[51] E Pennestrì and R Stefanelli. Linear algebra and numerical algorithms using dual numbers. *Multibody System Dynamics*, 18(3):323–344, 2007.

[52] Matthew Smith. Applications of dual quaternions in three dimensional transformation and interpolation. 2013.

[53] Ben Kenwright. A beginners guide to dual-quaternions: what they are, how they work, and how to use them for 3d character hierarchies. 2012.

[54] Ladislav Kavan, Steven Collins, Carol OSullivan, and Jiri Zara. Dual quaternions for rigid transformation blending. *Trinity College Dublin, Tech. Rep. TCD-CS-2006-46*, 2006.

[55] Ettore Pennestrì and Pier Valentini. Dual quaternions as a tool for rigid body motion analysis: A tutorial with an application to biomechanics. *Archive of Mechanical Engineering*, 57(2):187–205, 2010.

[56] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[57] Mohinder S Grewal. Kalman filtering. In *International Encyclopedia of Statistical Science*, pages 705–708. Springer, 2011.

[58] Robert Grover Brown and Patrick YC Hwang. Introduction to random signals and applied kalman filtering: with matlab exercises and solutions. *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions, by Brown, Robert Grover.; Hwang, Patrick YC New York: Wiley, c1997.*, 1997.

[59] Lennart Ljung. Asymptotic behavior of the extended kalman filter as a parameter estimator for linear systems. *IEEE Transactions on Automatic Control*, 24(1):36–50, 1979.

[60] Masaru Hoshiya and Etsuro Saito. Structural identification by extended kalman filter. *Journal of engineering mechanics*, 110(12):1757–1770, 1984.

[61] Young-Real Kim, Seung-Ki Sul, and Min-Ho Park. Speed sensorless vector control of induction motor using extended kalman filter. *IEEE Transactions on Industry Applications*, 30(5):1225–1233, 1994.

[62] João Luís Marins, Xiaoping Yun, Eric R Bachmann, Robert B McGhee, and Michael J Zyda. An extended kalman filter for quaternion-based orientation estimation using marg sensors. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 4, pages 2003–2011. IEEE, 2001.

[63] G Rigatos and S Tzafestas. Extended kalman filtering for fuzzy modelling and multi-sensor fusion. *Mathematical and computer modelling of dynamical systems*, 13(3):251–266, 2007.

[64] Leopoldo Jetto, Sauro Longhi, and Giuseppe Venturini. Development and experimental validation of an adaptive extended kalman filter for the localization of mobile robots. *IEEE Transactions on Robotics and Automation*, 15 (2):219–229, 1999.

[65] Fred Daum. Nonlinear filters: beyond the kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, 20(8):57–69, 2005.

[66] Vincent Lepetit, Pascal Fua, et al. Monocular model-based 3d tracking of rigid objects: A survey. *Foundations and Trends® in Computer Graphics and Vision*, 1(1):1–89, 2005.

[67] Benjamin Busam, Marco Esposito, Simon Che'Rose, Nassir Navab, and Benjamin Frisch. A stereo vision approach for cooperative robotic movement

therapy. In *IEEE International Conference on Computer Vision Workshop (ICCVW)*, December 2015.

[68] Joel Janai, Fatma Güney, Aseem Behl, and Andreas Geiger. Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. *arXiv preprint arXiv:1704.05519*, 2017.

[69] Sebastian OH Madgwick, Andrew JL Harrison, and Ravi Vaidyanathan. Estimation of imu and marg orientation using a gradient descent algorithm. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, pages 1–7. IEEE, 2011.

[70] xioTechnologies. Oscillatory-motion-tracking-with-x-imu. https://github.com/xioTechnologies/Oscillatory-Motion-Tracking-With-x-IMU, 2017.

[71] S. Garrido-Jurado R. Muoz-Salinas. Aruco library, 2013. URL https://sourceforge.net/projects/aruco/. [Online; accessed 1-April-2018].

[72] Junichi Tokuda, Gregory S Fischer, Xenophon Papademetris, Ziv Yaniv, Luis Ibanez, Patrick Cheng, Haiying Liu, Jack Blevins, Jumpei Arata, Alexandra J Golby, et al. Openigtlink: an open network protocol for image-guided therapy environment. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 5(4):423–434, 2009.