

# Multimodal sensor data fusion methods for infrastructureless head-worn interfaces

– Sensor systems for robust and adaptive human-robot collaboration –

Vom Fachbereich für Physik und Elektrotechnik  
der Universität Bremen

zur Erlangung des akademischen Grades

Doktor–Ingenieur (Dr.-Ing)

genehmigte Dissertation

von

M.Sc. Lukas Wöhle

wohnhaft in Bremen

Referent:

Prof. Dr.-Ing. Axel Gräser

Korreferent:

Prof. Dr.-Ing Walter Lang

Eingereicht am:

09.08.2021

Tag des Promotionskolloquiums:

31.01.2022



## Abstract

Human-robot collaboration is becoming increasingly important especially in the context of rehabilitation robotics where people use robots to regain autonomy. For this purpose, a variety of approaches to control these systems has been developed. A highly intuitive approach is head-motion based control, which enables precise mapping of 3D control commands onto a system via deliberate head movements.

This thesis presents a system to ensure the necessary robustness and adaptivity for the control of a robotic system by means of head motion. For that purpose, a lightweight, infrastructureless sensor system was developed that can be worn on the head to fully control a robotic system in all degrees of freedom in Cartesian space. The system is modular in design and data fusion scheme to grant as much adaptivity as possible. The core of the sensor system consists of a Magnetic, Angular Rate, and Gravity (MARG) sensors, which are used to determine the orientation of an object in 3D space. The orientation computation is based on the numerical integration of angular rate measurements from a three-axis gyroscope. Unfortunately Micro-Electromechanical Systems (MEMS) gyroscopes are subject to noise terms that degrade the orientation estimation. To counteract this, MARG sensors are equipped with global reference measurement sensors: an accelerometer and magnetometer. The accelerometer is used to correct orientation in the plane perpendicular to gravity, while the magnetometer is used as an electronic compass to correct the remaining axis. This arrangement enables a globally referenced orientation computation. However, magnetometers are subject to interference, which can completely invalidate its use as a reference measurement. To increase robustness against such disturbances, a data fusion process has been developed which compensates short-term disturbances and allows for simple incorporation of additional references for error correction without further effort. On this basis, a novel approach was developed that uses the physiological coupling of a human's eyes and head rotation to support the MARG sensor's orientation determination during long-term magnetic field perturbations. Experimental data demonstrates that this method provides an error reduction of up to 50 percent. The usage of an eye tracker logically opens up the use of visual methods for orientation determination. Therefore, within this thesis an open-source visual Simultaneous Localization And Mapping (SLAM) for RGB-D cameras is integrated into the data fusion process to enable a robust calculation of the head pose in space. The data fusion process is designed to dynamically switch between magnetic, inertial, eye tracking-based and visual reference technologies to enable robust orientation estimation under various perturbations, e.g. gyroscope bias, magnetic disturbances and visual sensor data failure. The combination of these sensors and methods provides the capability, in addition to sensing head rotation only, of precise eye or head gaze vector control to perform accurate positioning of a robot's End Effector (EEF) in Cartesian space.

The work is finalized with a functional verification of the system in a human-robot workplace, which indicates that the sensor system and methods enable a precise control mechanism for robot teleoperation.

## Kurzfassung

Die Mensch-Roboter-Kollaboration gewinnt zunehmend an Bedeutung, speziell im Kontext der Rehabilitationsrobotik, indem Menschen Roboter nutzen, um ein Stück der eigenen Autonomie wiederherzustellen. Hierzu sind vielfältige Ansätze zur Steuerung dieser Systeme entwickelt worden. Eine äußerst intuitive Möglichkeit bietet die Steuerung über Kopfbewegungen, durch die präzise 3D-Steuerbefehle auf ein System gemapped werden.

Diese Thesis präsentiert ein System, um die nötige Robustheit und Adaptivität der Steuerung robotischer Systeme mittels Kopfbewegungen zu gewährleisten. Dazu wurde ein leichtes, infrastrukturloses Sensorsystem entwickelt, das auf dem Kopf getragen werden kann und die Steuerung eines Roboters in allen Freiheitsgraden erlaubt. Das System sowie Datenfusion sind modular aufgebaut, um möglichst adaptiv eingesetzt werden zu können. Das Herzstück besteht aus einem MARG-Sensor, welcher zur Orientierungsermittlung im 3D Raum genutzt wird. Die Orientierungsberechnung basiert auf der numerischen Integration von Drehratenmessungen eines Gyroskops. Jedoch unterliegen MEMS-Gyroskope Rauschtermen, die die Lagebestimmung maßgeblich erschweren. Um der Fehlerakkumulation entgegenzuwirken, besitzen MARG's weitere Sensoren: einen Beschleunigungssensor und ein Magnetometer. Der Beschleunigungssensor dient zur Korrektur der Orientierung in der Ebene senkrecht zur Schwerkraft, während das Magnetometer als elektronischer Kompass zur Korrektur der verbliebenen Achse verwendet wird. Diese Anordnung ermöglicht eine global referenzierte Lagebestimmung. Allerdings sind auch Magnetometer störanfällig, was die Messung mitunter vollständig unbrauchbar machen kann. Um die magnetische Robustheit zu erhöhen, wurde ein Datenfusionsprozess entwickelt, welcher kurzzeitige Störungen kompensiert und die Möglichkeit der einfachen Integration weiterer Referenzen zur Fehlerkorrektur bietet. Auf dieser Basis wurde ein neuartiger Ansatz entwickelt, welcher die physiologische Kopplung von Augen- und Kopffrotation eines Menschen nutzt, um die Orientierungsermittlung des Sensors während lang anhaltender Magnetfeldstörungen zu unterstützen. Experimentelle Daten belegen, dass diese Methode eine Fehlerreduktion von bis zu 50 Prozent ermöglicht. Der Einsatz des Eyetrackers eröffnet als logische Konsequenz den Nutzen visueller Orientierungsermittlung. Daher wurde in dieser Thesis ein Open-Source vSLAM für RGB-D Kameras in den Datenfusionsprozess integriert, um eine robuste Berechnung der Kopfpose zu ermöglichen. Die Datenfusion ist so konzipiert, dass dynamisch zwischen magnetischen, inertialen, Eyetracking-basierten und visuellen Referenzen gewechselt wird, um eine robuste Orientierungsschätzung unter verschiedenen Störungen zu ermöglichen, u. a. magnetische Störungen oder der Ausfall visueller Daten. Das Gesamtsystem ermöglicht es, neben der Erfassung der Kopffrotation, eine präzise Augen- oder Kopfvektorsteuerung für die genaue Positionierung des Roboters im kartesischen Raum vorzunehmen. Die Arbeit wird mit der funktionalen Verifizierung des Systems in einem Mensch-Roboter-Arbeitsplatz finalisiert. Diese zeigt, dass das System und die Methoden einen präzisen Mechanismus für die Roboter-Teleoperation liefern.

---

## **Ehrenwörtliche Erklärung**

Hiermit versichere ich, die vorliegende Arbeit ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche gekennzeichnet. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Bremen, den 09.08.2021

---

Lukas Wöhle



# Contents

<b>List of Acronyms</b>	<b>ix</b>
<b>List of Symbols</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Related works . . . . .	2
1.2 Challenges . . . . .	6
1.3 Main contribution of this thesis . . . . .	8
1.4 Thesis outline . . . . .	9
<b>2 Sensor technologies for head-worn interfaces</b>	<b>11</b>
2.1 MARG sensors . . . . .	11
2.1.1 Working principle . . . . .	11
2.1.2 Noise sources . . . . .	17
2.1.3 MARG sensor data fusion principles . . . . .	36
2.2 Mobile eye tracker . . . . .	37
2.2.1 Mobile video-based eye tracking . . . . .	39
2.2.2 Video-based eye tracker noise sources . . . . .	41
2.3 Mathematical concepts of orientation representation . . . . .	42
2.3.1 Preliminary conventions . . . . .	42
2.3.2 Euler angles . . . . .	46
2.3.3 Unit-quaternion vector . . . . .	49
2.3.4 MARG sensor only orientation estimation methods . . . . .	51
2.3.5 Visual orientation estimation methods . . . . .	57
<b>3 Infrastructureless head orientation estimation</b>	<b>63</b>
3.1 General concept . . . . .	63
3.2 Sensor system setup and design . . . . .	67
3.2.1 Modular system design . . . . .	67
3.2.2 The wireless orientation tracking system . . . . .	69
3.2.3 Mobile eye tracker - Pupil Core . . . . .	76
<b>4 MARG sensor based head orientation estimation</b>	<b>79</b>
4.1 A quaternion-based filter framework . . . . .	80
4.1.1 Orientation from angular rate . . . . .	80
4.1.2 Orientation from the earth's magnetic and gravity field . . . . .	81

4.2	Robust quaternion-based Kalman filter . . . . .	85
4.3	Maintaining orientation under magnetic disturbance . . . . .	87
4.4	Setup and experimental results . . . . .	90
4.5	Conclusion . . . . .	94
<b>5</b>	<b>Eye tracking-supported MARG sensor orientation estimation</b>	<b>97</b>
5.1	SteadEYE-head: Using eye tracking data to improve heading estimates . . .	101
5.1.1	Visual zero-rotation detection . . . . .	102
5.1.2	MARG sensor data fusion . . . . .	103
5.2	Setup and experimental results . . . . .	108
5.2.1	Interchangeable north direction vector substitutes . . . . .	111
5.2.2	Magnetic disturbance detection . . . . .	113
5.2.3	MARG sensor data fusion approach using visual fixations . . . . .	115
5.3	Conclusion . . . . .	117
5.3.1	Limitations . . . . .	118
<b>6</b>	<b>Visual SLAM-supported MARG sensor orientation estimation</b>	<b>121</b>
6.1	Filter development . . . . .	122
6.1.1	Visual SLAM-based pose estimation . . . . .	123
6.1.2	Visual-inertial orientation fusion . . . . .	125
6.1.3	Visual position estimation . . . . .	128
6.2	Beyond head motion – 3D gaze-based robot control . . . . .	129
6.3	Setup and experimental results . . . . .	133
6.4	Visual-inertial orientation estimation accuracy . . . . .	135
6.5	Visual position estimation accuracy . . . . .	138
6.6	Three-dimensional gaze point estimation accuracy . . . . .	139
6.6.1	Head gaze . . . . .	140
6.6.2	Eye gaze . . . . .	140
6.7	Conclusion . . . . .	144
<b>7</b>	<b>Verification in a human-robot collaboration use case</b>	<b>145</b>
7.1	Use case – mechanical assembly of wedge adjuster . . . . .	145
7.2	Workplace setup . . . . .	146
7.2.1	Control strategy . . . . .	148
7.3	Results . . . . .	151
7.3.1	Control accuracy of Cartesian pointing . . . . .	151
7.3.2	Teleoperation – isolation and grasping . . . . .	154
<b>8</b>	<b>Discussion</b>	<b>157</b>
<b>9</b>	<b>Outlook</b>	<b>161</b>
	<b>Bibliography – authored and co-authored publications</b>	<b>165</b>

*Contents*

---

<b>Bibliography</b>	<b>174</b>
<b>List of figures</b>	<b>175</b>
<b>List of tables</b>	<b>178</b>



# List of Acronyms

**MARG** Magnetic, Angular Rate, and Gravity

**EEF** End Effector

**IR** Infrared

**ROI** Region Of Interest

**GUI** Graphical User Interface

**MIA** Mensch-Roboter Interaktion im Arbeitsleben bewegungscheingeschränkter Personen

**LiPo** Lithium-Ion Polymer

**CGI** Computer Generated Imagery

**ToF** Time-Of-Flight

**VO** Visual Odometry

**GDA** Gradient-Descent Algorithm

**RTOS** Real Time Operating System

**WiOT** Wireless Orientation Tracking System

**BLE** Bluetooth Low Energy

**UDP** User Defined Protocol

**LDO** Low Drop-out

**THT** Through-hole Technology

**SMT** Surface-mount Technology

**ISR** Interrupt Service Routine

**ZRU** Zero Rotation Update

**ID** Identifier

<b>SLAM</b>	Simultaneous Localization And Mapping
<b>NED</b>	North-East-Down
<b>CMOS</b>	Complementary Metal-Oxide-Semiconductor
<b>CCD</b>	Charge-Coupled-Device
<b>DoF</b>	Degrees Of Freedom
<b>ARW</b>	Angle Random Walk
<b>RRW</b>	Rate Random Walk
<b>PSD</b>	Power Spectral Density
<b>MEMS</b>	Micro-Electromechanical Systems
<b>IMU</b>	Inertial Measurement Unit
<b>VOR</b>	Vestibulo-Ocular Reflex
<b>DCM</b>	Direction Cosine Matrix
<b>BA</b>	Bundle Adjustment
<b>ORB</b>	Oriented FAST And Rotated BRIEF
<b>FOV</b>	Field Of View
<b>SLERP</b>	Spherical Linear Interpolation
<b>IPC</b>	Inter Process Communication
<b>API</b>	Application Programming Interface
<b>ROS</b>	Robot Operating System
<b>VIO</b>	Visual Inertial Odometry
<b>TCP</b>	Tool Center Point
<b>AR</b>	Augmented Reality
<b>VR</b>	Virtual Reality
<b>PID</b>	Proportional Integral Derivative
<b>IK</b>	Inverse Kinematics
<b>FK</b>	Forward Kinematics

# List of Symbols

$\vec{F}$	force
$m$	mass
$\vec{a}$	acceleration
$\vec{f}$	specific force
$\vec{g}$	gravity
$\vec{m}$	magnetic field
$\vec{v}$	velocity
$\vec{\omega}$	angular velocity \ angular rate
$\vec{B}$	magnetic field
$\vec{E}$	electrical field
$U$	voltage
$I$	current
$t$	time
$\omega_0$	resonance frequency
$k_B$	Boltzmann constant
$\Omega_N$	noise equivalent angular rate
$a_N$	noise equivalent acceleration
$T$	temperature
$\vec{b}_{xx}$	bias vector of sensor xx
$\vec{n}_{xx}$	inherent noise vector of sensor xx
$M_{xx}$	misalignment matrix of sensor xx
${}^N\vec{x}\vec{x}$	vector xx w.r.t. navigation coordinate system
${}^B\vec{x}\vec{x}$	vector xx w.r.t. body coordinate system
${}^E\vec{x}\vec{x}$	vector xx w.r.t. earth coordinate system
${}^I\vec{x}\vec{x}$	vector xx w.r.t. inertial coordinate system
$\sigma_{xx}$	standard deviation of variable xx
$\sigma_{xx}^2$	variance of variable xx
$\mathcal{N}(\mu, \sigma^2)$	normal distribution with mean $\mu$
$\sigma(\tau)$	Allan deviation
$\sigma^2(\tau)$	Allan variance
$\tau$	cluster duration
$S(f)$	power spectral density
$f$	frequency
$\phi, \theta, \psi$	Euler angles - roll, pitch, yaw

---

${}^N_B\mathbf{q}$	quaternion denoting rotation from body to navigation coordinate system
${}^N_B\dot{\mathbf{q}}$	conjugate quaternion
${}^N_B\dot{\mathbf{q}}$	quaternion derivative
${}^N_B C$	rotation matrix denoting rotation from body to navigation coordinate system
${}^N_B T$	transformation matrix (rotation and translation)
$\vec{x}_k$	state vector
$\vec{y}_k$	innovation vector
$\vec{z}_k$	measurement vector
$\vec{u}_k$	control-input vector
$\vec{v}_k$	measurement noise vector
$\vec{w}_k$	system noise
$F_k$	state-transition matrix
$B_k$	control-input model matrix
$H_k$	measurement model matrix
$K_k$	Kalman gain matrix
$P_k$	state estimate covariance matrix
$R_k$	measurement noise covariance matrix
$Q_k$	system model noise covariance matrix
$I_{n \times n}$	identity matrix
$J$	Jacobian matrix
$\nabla f$	gradient
$\Sigma_{xx}$	covariance matrix of sensor xx
$c_b$	convergence value
$\alpha$	gain factor
$\epsilon_{xx}$	threshold

# 1 Introduction

The collaboration of humans and robots is becoming an increasingly important field of research within various fields. For example in the context of "Industry 4.0" [8], where human-robot workplaces in which direct collaboration between humans and robots takes place will shape the future of tomorrow's industry. A further field of application and research is the broad area of human-computer interfaces for motor impaired people, e.g. assistive or rehabilitation robotics, due the increasing number of people who require assistive technologies to perform activities of daily living [9]. Especially the latter field has gained increased research attention in the last decades to enable people with physical disabilities to regain a certain amount of autonomy either in their working or daily life [10]. However, in contrast to current human-robot collaborations where a human and robot work in isolation separated by a fence, future applications will feature direct interaction side by side [11]. This interaction may range from teleoperation, where the human controls the robot and is in charge of decision-making, to supervisory control where the robot works mostly autonomous overseen by a human supervisor [12]. This thesis focuses mostly on teleoperation due to the following reasons: a) the current visual superiority and context-based decision-making of the human user, b) the human's ability to fast and near unrestricted planning from observations and c) no legal issues regarding liability since the robot is obeying to the human's command [13].

The afore mentioned direct collaboration in these workplaces demands a high level of robustness to enable direct interaction between humans and robots. This point is crucial in the context of rehabilitation robotics since a motor-impaired person is even more vulnerable to a possible robot failure or malfunction.

Since humans and robots are expected to collaborate side by side, the human user needs input or communication modalities to communicate his or her intention or directly control (steer) the robot to perform a certain task. Especially in situations where the hands of a person are either occupied (Industry 4.0) or not usable, e.g. people with severe physical disabilities, a safe, reliable and intuitive communication source that enables a direct interaction with the robotic system should be provided. Such an interface provides the opportunity for people suffering from physical motor impairments to be (re)integrated into the working life, e.g., for pick-and-place tasks in a library workplace designed for people with tetraplegia [14]. The question of how an interface could be designed to effectively and intuitively allow for hands-free robot control has drawn significant research attention in the last decade [9, 15, 16, 17, 18]. Recent approaches focus on the use of head motion or eye gaze tracking data to allow for direct robot control since both

modalities are naturally correlated with direct interaction intention and enable accurate control mechanisms [19]. The head movements of a human can be measured using low-cost and lightweight MEMS MARG or Inertial Measurement Unit (IMU) sensors [17]. This form of input modality can also be used by motor-impaired people (with existing head mobility) to control assistive systems or robots and therefore increase autonomy of the user.

MARG sensors are miniature MEMS sensors that enable a three-dimensional orientation computation. The orientation computation is based on the numerical integration of angular rate measurements from a three-axis gyroscope. Unfortunately, MEMS based gyroscopes are subject to noise terms that degrades the orientation estimation. One noise term is the so-called DC bias, a slow changing offset, that leads to an accumulation of orientation errors (drift) from the integration step. To oppose this error, MARG sensors are equipped with a three-axis accelerometer and magnetometer. The accelerometer is used to measure the direction of the earth gravity field, which can be used to correct the orientation in at least two axes (the so-called pitch and roll axis). The magnetometer on the other hand is used as an electronic compass to correct the remaining axis (yaw axis). This set of sensors enables the calculation of a global referenced orientation estimation in three-dimensional space. However, accelerometers and magnetometers are also affected by noise, some of these can, in principle, render a measurement completely unusable.

This is the case for magnetic disturbances. These are, above all, the main source of interference, if the objective is a robust and error-free orientation calculation. These disturbances interfere with the determination of the reference vector (magnetic north pole) which is used to correct the relative orientation estimations from gyroscopes. This results in a deviation of the orientation estimation and can lead to a false control command to the robot and might make it difficult or even completely impossible to perform the task.

The goal of this thesis is to generate redundant, robust, and accurate data for specific interaction tasks to enable adaptive and robust human-robot interaction for future workplaces. To achieve this, new sensor and multi-sensor data fusion methods have been developed to ensure robust, hands-free and infrastructureless robot control.

## 1.1 Related works

Recent Interfaces focus on the use of head motion or eye tracking data to allow for continuous robot control and deliver a low-cost and intuitive control mechanism [16] [20]. Head motion-based interfaces usually employ low-cost sensors, i.e., MARG sensors, to estimate orientation without a need for static infrastructure, e.g., markers that would limit the useable motion range and environment [21].

During her PhD Thesis N. Rudigkeit developed an interface for direct head motion-based robot teleoperation, namely the AMiCuS System. The system only relies on MARG sensor-based orientation estimation. A MARG sensor is used to measure the three Degrees Of Freedom (DoF) head orientation which is mapped onto the robot's EEF. A change of the head angle is translated into a continuous and iterative change of the translation or rotation of the end effector. The amplitude of the deflection of the angles scales the speed of the iterative change. The head-angles are furthermore mapped onto a screen-based cursor application to switch between motion groups using dwell buttons. The separation of the control into motion groups solves the underrepresentation of the input to the possible output variables (3-DoF translation, 3-DoF rotation, gripper, etc.). This interface enables precise and intuitive robot teleoperation but requires a monitor and GUI application for robot control. The orientation estimation from MARG sensors is based on the angular rate integration measured by the gyroscope, which suffers from noise, especially gyroscope offset, that results in drift of the orientation estimation and therefore reduces accuracy. The drift is usually compensated by using global reference vector measurements from accelerometer and magnetometer [1] [22]. Since the magnetic field measurements are unstable in indoor environments and nearby the robot, robustness of the control signal cannot be guaranteed when using MARG sensors only. The orientation estimation tends to drift, especially when dealing with low-cost consumer-grade sensors.

Some industrial-grade commercially available MARG sensor systems offer strategies to enable a robust orientation estimation in magnetically challenging environments, e.g. XSens MTi series. This MARG sensor has a built-in feature called active heading stabilization (AHS) to enable a low drift unreferenced yaw orientation estimation. The company XSens states that this feature is able to deliver yaw drift as low as 1-5 degrees per hour, depending on the application [23]. According to XSens, it is not recommended to use AHS "when the application expects that either the MTi (MARG) or the magnetic field rotates very slowly" [24], since this might lead to detrimental effects on the yaw estimation. Furthermore XSens continues to state that the feature is best used within applications featuring "occasionally motionless phases, e.g. warehouse robotics or ground vehicles", [24]. For head motion-based robot control, these criteria, angular rates above a certain threshold, magnetic fields that do not change slowly and motionless (stationary) phases, cannot be guaranteed and might therefore not be well suited.

Caruso et al. recently published a study comparing ten different algorithms for MARG sensor data fusion and found that no statistical difference is evident when the filters are optimally tuned. They found, that high angular rates typically lead to an increase in estimation errors for all filters. Furthermore, the accuracy mainly depends on the quality of the MARG sensors' hardware. The highest accuracy can be achieved using high-performance MARG sensors (e.g. XSens Units) [25], which come at high price ranges (starting at > 400 euros). Nevertheless, if magnetic disturbances are present, any orientation estimation method using MARG sensors only reaches its limitations and will

result in accumulation of errors with respect to time.

Besides orientation, MARG sensors can be used to estimate position, at least for a certain period of time. Centimeter-accurate position estimation with MARG sensors, however, is typically based on data fusion with external sensors (e.g., GPS) [23]. In GPS-denied environments (indoor applications) dead reckoning based on MARG sensors only can be used. This method relies on double integration of the acceleration to extract velocity and position. The double integration step will accumulate every minuscule error and position accuracy decreases rapidly (quadratic error growth). Depending on the desired application, orientation and position estimation accuracy based on a single sensing MARG sensor might be sufficient. In terms of specifications for the use case of a human-robot collaboration scenario, these errors have to be minimized. This is especially true for 3D gaze-point estimation in Cartesian space. A precise position and orientation estimation must be provided to calculate an accurate gaze position in Cartesian space. Every head pose error will directly influence the gaze point prediction. Therefore, other heading reference sources must be provided to account for the orientation and position drift. Furthermore, using head motion only shows the shortcoming of having to switch between various motion groups to map from the underrepresented 3D motion space of the head to the full 6D (3-DoF rotation, 3-DoF translation) EEF motion range.

eye gaze-based robot control utilizes the natural eye motion of a human. A recent approach utilizes 2D gaze points and gaze gestures to control the pose of a robotic arm in three-dimensional space [20]. The user can switch between motion groups by gaze gestures to control the arm in various directions (xy-plane, z-plane, rotational xy, rotational z) and control the robot by gazing at a dynamic command area (Region Of Interest (ROI)) at the center of the robots' gripper. This interface needs to track the robots EEF position in order to specify the dynamic command areas and therefore needs a fiducial marker at the EEF, so it relies on infrastructure in terms of stationary markers in the real-world scenario. Furthermore, the interface needs more motion group transitions since it only generates two-dimensional commands for a six-dimensional control problem.

In [2] Stalljann et al. researched head motion and eye gaze-based performance for robot teleoperation using a Graphical User Interface (GUI)-based control approach presented in [26]. They found that MARG sensor-based cursor control is less error-prone compared to eye gaze-based cursor control and that eye gaze control is more mentally demanding. Furthermore, a multimodal control approach was presented, combining eye gaze and head motion to control a robotic arm for a drinking task. The eye gaze was used to control the cursor to switch between motion groups while the head motion was used to control the robotic arm. Even though every participant (including an individual with tetraplegia) was able to perform the robot-assisted drinking task, "switching between the head control and eye control was not easy" for the participants [2]. This is mainly due to slippage of the eye tracking glasses and eye gaze calibration errors, resulting in an offset of the gaze position on the screen. This indicates that a pure eye gaze-based

cursor control approach might not be an ideal modality for continuous control signals. The interface furthermore only utilizes MARG sensors for head motion orientation measurements and as such will experience DC bias-induced drifts.

Tostado et al. [27] proposed an interface decoding eye gaze into a position in 3D space for robotic end-point control. The proposed system is capable of controlling a robotic end effector in 3D space by directly gazing at a specific object or point. The system consists of two stationary eye cameras in front of the user and thus is dependent on the infrastructure which limits the possible workspace related to the cameras' field of view and motion range. Furthermore, the system does not include head tracking capabilities and therefore assumes a fixed head position, which is ensured by a chin rest that further reduces mobility.

Scalera et al. [28] present a robotic system that enables robotic painting by eye gaze. The work addresses the prior mentioned mobility restrictions by using a stationary eye tracking bar (Tobii Eye Tracker 4C) that is attached to a computer screen in front of the user. The eye tracking bar accounts for head motions of the user and therefore makes accurate eye gaze tracking on a computer screen possible, enabling a less restrictive head motion policy, i.e., no chin rest. The filtered eye gaze coordinates on the computer screen are the input coordinates for the Tool Center Point (TCP) position of a robot on a painting canvas. This approach eliminates most mobility restrictions but relies on the stationary eye tracking camera attached to a computer screen and only supports two-dimensional commands in a single plane from the computer screen mapped onto the robot's workspace. A further pitfall of this approach occurs if multiple persons are in the field of view of the tracker, ultimately disturbing the head motion estimation pipeline and therefore eye tracking accuracy.

Dezmien et al. [29] developed an interface for Eye gaze-based robotic writing and drawing. Similarly to Scalera et al. [28], this work utilizes a stationary eye tracking bar on a surface (Tobii Eye X) to track eye gaze-points of a user on a 2D canvas to directly control a robotic arm drawing on the canvas. The approach uses the Tobii Eye X's head tracking capability to attach or detach the pen on the canvas. Likewise to the prior mentioned approach, the interface enables direct low cost eye gaze robot control but relies on the stationary camera hardware and is applicable only in a two-dimensional plane.

During the research for this thesis, a data fusion process was developed incorporating a head-worn MARG sensor and the afore mentioned stationary eye tracking bar (Tobii Eye Tracker 4C) to enable robust orientation estimation while long-term magnetic disturbances are present [3]. This work is based on a filter framework that is presented in detail in this thesis. The data fusion involves a delta quaternion fusion (rotational differences between two quaternions) between the MARG sensor and the computer vision-based head tracking of the stationary eye tracking bar. A state machine is used to decide whether to incorporate the head orientation estimation from the video-based head tracking or switch to magnetometer measurements or discard both. This application enables robust orientation estimation and solves the problem that occurs from the video-based

head tracking if more than one person is in the field of view of the camera, since the MARG sensor is used as a redundant measurement. Nevertheless, the approach limits the available motion space and is by no means infrastructureless or self-contained.

A recent approach delivers a potential interface for a full eye gaze-based control of a robotic arm by combining eye tracking, head tracking and depth information [30]. This interface, however, is dependent on a stationary infrared motion tracking system for head pose estimation. The motion capture system cannot be used in mobile applications and furthermore exceeds reasonable costs for a control interface. The control approach only includes three-dimensional EEF position control and does not include end effector orientation. It utilizes the human operators hand as the end effector rather than a robotic gripper, solving the problem of missing EEF control. The operator's hand is coupled to the robotic arm by a magnetic coupling and therefore can only be used by people that are able to close their hands and grab an object.

The HoloLens 1 is a commercially available interface that is capable of delivering accurate three-dimensional head gaze but lacks the ability to track eye positions and deliver eye gaze. The HoloLens 1 is weighing 579 *g* in total [31]. The center of mass is at the front of the head and might therefore be too heavy to be used for a long period of time by people with severe physical disabilities. The new generation of the HoloLens, the HoloLens 2, is able to deliver eye and head gaze vectors and comes at a total weight of 566 *g* distributed more equally [32]. Currently, it has yet to be researched if the HoloLens 2 is more suitable for people with severe physical disabilities in long-term use.

## 1.2 Challenges

Head-worn interfaces can enable hands-free interaction and control of robotic systems of any kind. To ensure safety and robustness in all situations needed, these interfaces need to generate reliable control signals. Head motion-based interfaces have proven to deliver an intuitive robot control method. A single consumer-grade MEMS MARG sensor can be used for low cost ( $< 1$  Euro) and lightweight infrastructureless head motion measurements, since the sensor enables an orientation computation with respect to global quantities, i.e., gravity and geomagnetic field, by combining angular rate measurements and correcting drift using the global references. In an ideal scenario, a MARG sensor robot control interface therefore is in no need for additional local requirements, e.g. stationary cameras, tracking markers and so on.

Unfortunately, real-world applications push MARG sensors and their usage as the only sensing technology for accurate orientation estimation to their limits. The performance of a MARG sensor depends on an abundance of potential noise sources as well as the dynamics of the head motion that is to be measured. These disturbances directly affect the measured sensor data and thus the quality of the orientation estimation. This is particularly true for the gyroscope's internal DC bias noise effects as well as the surrounding

magnetic environmental conditions. MEMS gyroscopes are subject to a slow varying DC offset (bias) embedded in the angular rate measurement. Upon integration of angular rate to estimate the current angle, the bias will lead to an accumulation of errors. Even at rest, this bias will lead to a drift of the angle from its true value. This offset is usually corrected using the global vector quantities, gravity for roll and pitch components and the geomagnetic field to correct the yaw estimation. However, the environmental conditions in indoor environments generate magnetic disturbances that massively impact the usage of the geomagnetic field to correct this heading drift accumulation. Whether it is possible to use a magnetometer at all depends on the magnetic environment [33]. Since the magnetic environment for robotic collaboration is utmost unstable, especially in case of indoor usage and nearby the robot, accurate orientation estimation cannot be guaranteed when using MARG sensors only. The particular challenge, above all, lies within an environment dominated by permanent and ferromagnetic sources or materials that change the direction and/or magnitude of the measurement of the geomagnetic field at the direct location of the MARG sensor. Persisting (long-term) magnetic disturbances cannot be compensated by briefly switching off the magnetometer data. Wong et al. conclude that there is no efficient way of dealing with continuously changing magnetic field disturbances over long periods of time [34], at least if the problem is not constrained somehow (i.e. relative orientation from multiple IMUs, angular rate clipping). If a magnetic field disturbance persists, the DC bias of the gyroscope during the integration leads to continuous drift in the orientation solution that cannot be corrected, since the reference vector (geomagnetic field) is massively falsified. In order to prevent or significantly reduce this drift, appropriate procedures are researched to counteract long-lasting magnetic field disturbances and thus enable an orientation estimation without the need for local infrastructure. Apart from this, use case conditions of a head-worn interface do limit the usage of some well-established methods that enable MARG sensor orientation estimation without magnetometers. It is possible to use multiple accelerometers on two rigid bodies connected over a joint (e.g. upper and lower arm) to estimate relative orientation if the bodies are accelerated. Since the two IMUs measure this acceleration on different axes [35], it is possible to retrieve the relative orientation. Unfortunately, in the scenario considered here, neither a sufficiently high acceleration can be performed over several body segments, nor is the peer group capable of generating these accelerations. A user would have to wear an IMU on the head as well as the torso and perform a common acceleration of the body (e.g. forward) in order to enable the calculation of a relative pose.

Developing new functional combinations of multimodal sensing technologies can fill this gap and enable robust orientation estimations. The main challenge is to find new ways of data fusion of functional combination methods that do not or only slightly depend on local infrastructure, e.g. not involving special markers, stationary cameras, to maintain a self-contained solution that is not bound to special environmental conditions and does not limit the usable motion space. The particular challenge lies mainly in the research of

novel and unknown dependencies between the sensor signals and modalities measured (e.g. combining eye motion and head motion measurements).

### 1.3 Main contribution of this thesis

The objective of this thesis is the development of data fusion methods for multimodal infrastructureless sensor systems enabling robust robot control by means of head motion as well as head and eye gaze. The methods should provide a foundation for robust head pose estimation in three-dimensional space using low-cost consumer-grade MEMS MARG sensors as the system's core.

MARG sensors are highly monolithically integrated systems that enable the computation of an orientation estimation with respect to the gravitational and geomagnetic field. These sensors are subject to inherent DC bias and magnetic disturbances which interfere with the measurement of the reference vector (magnetic north pole) and lead to an accumulation of heading drift. To provide a precise orientation estimation for safe robot teleoperation, a robust data fusion method is introduced which reduces the influence of short time magnetic disturbances on MARG sensors. The filter combines a gradient descent method alongside a linear Kalman filter to compute a continuous orientation estimation represented as a quaternion. The gradient descent function provides an input measurement as a complete quaternion from vector observations for the Kalman filter. This is to a) enable a motion-weighted orientation estimation, b) bypassing the gimbal lock problem if dealing with a Euler angles-based orientation representation and c) open up the possibility to use further sources of vector observations to form a complete quaternion inside the gradient descent filter stage without the need of adapting the Kalman filter's set of equations. Based on this foundation, this thesis exploits and depicts methods for new functional data fusion combinations of MARG sensors and common head-worn sensing modalities to counter inherent sensor drift, i.e. using visual fixations from video-based eye tracking for orientation estimation enhancements. With the recent technological advances in camera miniaturization, MARG sensor estimation accuracy, mobile eye tracking and computer vision, this thesis provides methods alongside an infrastructureless and lightweight solution for accurate head motion as well as head and eye gaze-based 6-DoF robot control in Cartesian space to facilitate hands-free and multimodal human-robot collaboration without the need for an additional GUI. The system is composed of a three-part methodically linked data fusion approach that combines MARG sensor-based orientation measurements with eye tracking data as well as vision-based orientation estimations to produce accurate and robust orientation and position estimations. The methods are logically based on each other which gives the system the ability to fall back towards a sensing technology solely relying on MARG sensors if any other technology fails. The system does not rely on stationary cameras and is therefore infrastructureless and mobile regarding potential operational environments and usable motion space.

## 1.4 Thesis outline

The thesis is divided into nine chapters. Chapter 2 covers the theoretical background of common sensing technologies for head-worn interfaces, emphasizing the specific technologies employed here. It covers the basics of MARG sensor and mobile eye tracking, especially their working principles and noise sources. The chapter furthermore outlines mathematical concepts of orientation representation and an overview of current data fusion methods for MARG or visual pose estimation.

Chapter 3 presents the general concept of the proposed data fusion methods. These methods are verified based on a multimodal sensor system that was developed during this thesis. The chapter highlights the hard- and software components of the sensor system. These are a self-designed MARG sensor board (the Wireless Orientation Tracking System (WiOT) system) as well as an open-source, low-cost and lightweight eye tracker (the pupil core system).

Chapter 4 enfold the theory of the developed core data fusion framework for MARG sensor-based orientation estimation. This chapter contains the mathematical formulation of gradient descent as well as the linear Kalman filter and furthermore reports the experimental results and verification of the filter compared to a commercially available MARG sensor under normal and magnetically disturbed conditions. The content of the chapter is based on a peer reviewed conference paper [1] and is extended by further details.

In chapter 5 a novel method for eye tracking supported MARG sensor orientation estimation is introduced. The chapter emphasizes the physiological interconnection of head and eye motion and the concept for visual fixation-based zero orientation updates that can be employed in any type of MARG sensor data fusion process. The chapter presents the performance of the data fusion process verified against an infrared marker-based motion capture system under magnetically disturbed conditions. It outlines the performance increase by the functional combination of both sensing technologies without the need for external infrastructure. The content of the chapter is based on a peer reviewed journal paper [4] and is extended by further mathematical concepts, experimental data and concluding remarks.

Chapter 6 presents the development and results of a visual SLAM supported extension of the core filter framework to increase robustness and enable three-dimensional gaze point robot teleoperation. The chapter outlines a vector observation switching mechanism for the gradient descent stage that enables accurate orientation estimation from either magnetic, eye tracking or visual SLAM supported vector measurements. Finally, the chapter presents results of the interface for head pose estimation accuracy as well as head or eye gaze-point precision in Cartesian space. The content of the chapter is based on a peer reviewed journal paper [5], also extended by further details.

Chapter 7 presents a functional evaluation of the developed sensor system and methods in a human-robot collaboration scenario. As part of the BMBF-funded Mensch-Roboter

Interaktion im Arbeitsleben bewegungscheingeschränkter Personen (MIA)<sup>1</sup> project, providing the funding for the PhD position, a mechanical assembly use case was defined. Alongside the use case, a requirement profile was created and specifications were derived in cooperation with the project partner, "Büngern Technik", a sheltered workshop near Bocholt. The use case involves the assembly of a wedge adjuster (Keilsteller) by a human and a robot working together in a confined space at the same time. The control concept of the interface is based on the work of N. Rudigkeit from [26] and extended by Cartesian point control and eye-state-based control mode switching which enables robot control without the need for any GUI.

Finally, chapter 8 critically discusses the research conducted in this thesis, while chapter 9 outlines possible improvements and directions for future research.

---

<sup>1</sup>Federal Ministry of Education and Research of Germany grant number: 13FH011IX6

## 2 Sensor technologies for head-worn interfaces

This chapter presents the theoretical background and state-of-the-art regarding head-worn sensor technologies, focused on MARG sensors and video-based sensors for orientation estimation. The first section deals with MARG sensor hardware, the working principles, noise sources and general data fusion for orientation estimation using these sensors. This is followed by a section presenting the theoretical background for head-worn eye tracking systems, video-based eye tracking methods as well as potential noise sources. Furthermore, the chapter gives the mathematical background for orientation estimation and finalizes the chapter with an overview about principle and state of-the-art data fusion methods for MARG sensor and video-based orientation estimation principles.

### 2.1 MARG sensors

In general, a Magnetic, Angular Rate, and Gravity (MARG) sensor combines three different sensor triads to enable infrastructureless orientation estimation in three-dimensional space. As the name suggests, MARG sensors measure angular rate, gravity, and the geomagnetic field of the earth. These measurements are taken using a tri-axis gyroscope, a tri-axis accelerometer, and a tri-axis magnetometer [36]. If such a sensor does not feature a built-in magnetometer, it is termed IMU. The measurement principles and internal structure of the individual MEMS triads are described in detail below. MARG or IMU sensors are used in a wide variety of applications due to their low cost and exceptional small size. Nearly every smartphone nowadays features at least an IMU, in order to play motion-based mobile games, enable navigation in GPS-denied situations (driving through a tunnel) or image stabilization [37]. Furthermore, MARG sensors are used in aerospace [38], robotics [39], human robot interaction [17] and motion capture [36, 35] to create Computer Generated Imagery (CGI) characters or to characterize "bad" poses during manual labor and much more [35].

#### 2.1.1 Working principle

Generally speaking, a MARG sensor describes the meteorological quantities that such a sensor system can measure. This work utilizes microelectromechanical systems (MEMS) MARG sensors, due to their small size, low price, and availability. MEMS sensors are

highly monolithic integrated systems that include the sensing element or sensor, power supply circuit, readout and digitization circuits and furthermore a digital signal processing pipeline, e.g. different digital filters, noise suppression or even calibration routines. The following subsections briefly explain the physical quantity of measurement and the measurement principles for MEMS-based accelerometers, gyroscopes and magnetometers.

### Accelerometer

In theory, an accelerometer is a device to measure the rate of change of velocity of an object. This derivative, termed acceleration ( $\vec{a}$ ), is caused by a distinctive force, the Newton force  $\vec{F}$ , which is proportional to the inertial mass that is accelerated,

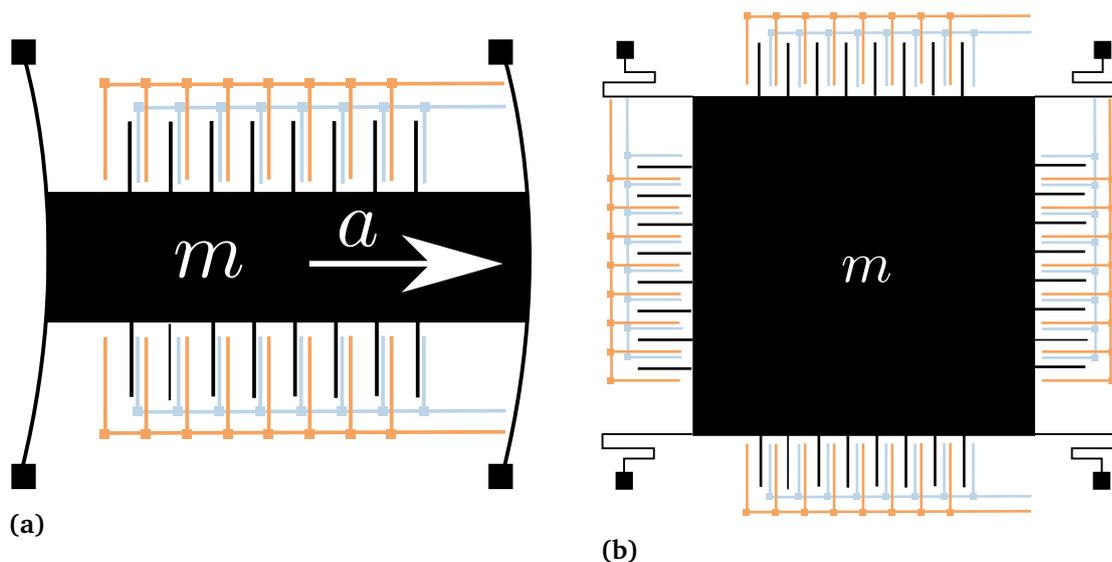
$$\vec{F} = m \cdot \vec{a}, \quad (2.1)$$

The SI unit of acceleration is  $m/s^2$  (meters/second<sup>2</sup>). Inertial-based accelerometers measure this force based on a mass spring system. The mass is subject to the Newton force when accelerated and will prolong or compress the spring. Measuring the deformation of the spring, or the deflection of the mass over time enables the calculation of the acceleration. A spring mass system needs damping to counter overshoot due to its second order time behavior.

Modern MEMS accelerometers measure the displacement of a spring-loaded mass by differential capacitive sensing between a reference frame and the spring-loaded mass. If an acceleration is applied to the seismic mass, the Newton force will deflect the mass and deform the spring causing a change in the distance between the two capacitive structures effectively changing its capacitance, which is proportional to the acceleration.

The crucial factor for high sensitivity of an accelerometer is the accurate measurement of the capacitance, which is the equivalent to the displacement that enables the calculation of acceleration. To enhance robustness, reduce nonlinearity and reject noise, the sensor is set up as a differential capacitance. The seismic mass is equipped with fingers (comb-like) that are embedded into a reference frame with two different potential counter fingers for each moving finger, such that each finger group (three) forms a differential capacitor which will undergo the same change of capacitance in opposite direction upon acceleration. Figure 2.1.1 depicts the schematic layout of a one- and two-axis MEMS accelerometer. To enlarge the dynamic range of a MEMS accelerometer a force feedback can be implemented which effectively maintains the position of the proof mass near zero by applying a voltage to the mass that generates an electrostatic force to counter the Newton force [40].

Since acceleration is a directional vector quantity, a set of three distinctive acceleration sensors is needed. This is either achieved by three single-axis MEMS accelerometers or one single- and one dual- axis accelerometer that are monolithically integrated onto a



**Figure 2.1:** Schematic of MEMS-based accelerometers - a) A one-axis MEMS accelerometer and b) a two-axis MEMS accelerometer. Every single finger of the spring-loaded seismic mass forms a differential capacitor to reject common electrical noise, adapted from [40].

single die. The measurement quantity of a MEMS accelerometer is termed specific force, abbreviated as  ${}^B \vec{f}$  [41]. This terminology puts emphasis on the fact, that a MEMS-based accelerometer measures a linear combination of every acceleration of the device. This means it measures a combination of the gravitational acceleration  ${}^B \vec{g}$  of the earth at all times as well as the linear introduced acceleration  ${}^B \vec{a}$  of the MEMS sensor or object it is attached to,

$${}^B \vec{f} = {}^B \vec{a} - {}^B \vec{g}. \quad (2.2)$$

To get the linear acceleration component of the body or object, in order to propagate the velocity or position change over time, the gravitational acceleration needs to be compensated. This compensation is typically implemented in the strapdown algorithm and essentially tries to predict the gravitational acceleration direction from the sensors orientation to isolate the body's acceleration [41]. Section 2.1.3 gives more details about this procedure.

## Gyroscopes

In terms of sensors, a gyroscope is a device used to measure angular velocity or the rate of rotation of an object. The earliest gyroscopes consist of a fast-rotating mass suspended within a gimbal-like structure. If the structure is rotated, the orientation of the spinning axis is unaffected following the law of conservation of angular momentum. These rotating gyroscopes are subject to imperfections due to their mechanical nature (e.g. bearing

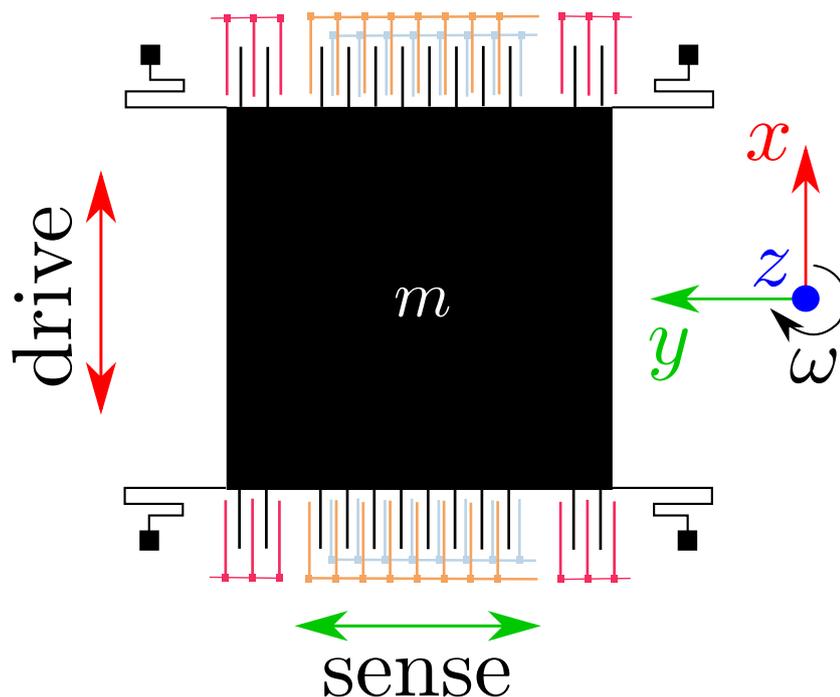
frictions), which is why other types of gyroscopes have been researched that replaced spinning wheel gyroscopes, e.g. mechanical vibrating structures or high-performance optical gyroscopes [42]. Modern MEMS gyroscopes are based on a vibrating or oscillating micro-mechanical structure that consists of capacitive structures. If this oscillating structure, moving at a velocity  $\vec{v}$ , is rotated at an angular velocity  $\vec{\omega}$  perpendicular to the motion, the Coriolis force acts perpendicular to the oscillating direction of the system leading to a deflection of the mass. If the rotation acts parallel to the linear motion, it vanishes [40]

$$\vec{F}_c = 2m\vec{v} \times \vec{\omega}. \quad (2.3)$$

The SI unit of angular velocity is radians per second, since radians are unit-less the SI unit can be written as  $s^{-1}$ . By definition, a clockwise rotation results in a negative angular velocity while a counterclockwise rotation has a positive angular velocity sign. The force  $F_c$  is linear to the magnitude of the applied angular velocity and is measured via capacitive changes between a driving frame, creating the oscillatory motion, and the sensing frame, which is essentially a MEMS accelerometer, and as such a seismic mass that is deflected due to the Coriolis force acting on it and accelerating it. The driving frame is based on the electrostatic actuation and consists of a comb-like capacitive structure, oscillating a spring-loaded mass in one axis. The sensing frame consists of a differential capacitance, likewise to a MEMS accelerometer, measuring the capacitive changes due to the Coriolis-force-induced acceleration of the seismic mass [40]. Figure 2.2 depicts a schematic of a single-axis MEMS gyroscope working principle.

Measuring small angular velocities using MEMS gyroscopes heavily depends on the decoupling of motion between the driving and the perpendicular sensing frame. W. Lang [40] gives a theoretical calculation for a gyroscope where the amplitude of the motion of the driving frame in x-direction is around  $5 \mu m$ , which is 500.000 times more compared to the amplitude of displacement of the y-axis ( $0.01 nm$  due to the Coriolis force from a  $0.1^\circ$  external rotation). This shows the importance of decoupling between the frames, (as) every minuscule misalignment will degrade the sensitivity. A small fraction of motion in the y-axis from the driving frame will ultimately result in a large error in the sensing frame. This fact highlights the difficulties and challenges in the designing process of MEMS gyroscopes. The reader is therefore referred to the relevant literature on this topic [40] [42].

A three-axis gyroscope is made from three single- (or multiple-) axis gyroscope, that are monolithically integrated onto one die. Using such a gyroscope enables the three-dimensional detection of angular velocity, which makes it possible to determine orientation changes by numerically integrating the angular velocity over time. MEMS gyroscopes are exceptionally well-suited to measure fast orientation changes of dynamic motions. They are commonly used for navigation and stabilization purposes in drones, smartphones, robotic and autonomous vehicle systems. Gyroscopes measure the rate of change and enable the determination of a relative orientation change. To estimate abso-



**Figure 2.2:** Schematic of MEMS-based gyroscope, adapted from [40]. The mass is oscillated through the electro-static actuators (red) in the x-axis direction. The sense differential capacitors measure a capacitance change due to the acceleration of the seismic mass from the Coriolis force along the y-axis (green) if an angular velocity  $\omega$  acts on the system perpendicular to the drive direction in the z-axis (blue).

lute orientation, the initial orientation of the system with respect to the navigation frame has to be known prior to the numerical integration process. Likewise to all physical measurement systems, gyroscopes are subject to noise, which leads to an accumulation of errors in the integration operation. This inevitably results in drift regarding the orientation estimation over time. Details of noise sources and their effects are described in section 2.1.2. To counteract this error accumulation, additional global references can be used to a) estimate the initial orientation and b) correct accumulations of errors.

### Magnetometer

A magnetometer is a device to measure the strength and direction of a magnetic field. A magnetic field  $\vec{B}$  is a vector field describing the effects of magnetic force on a moving electric charge/charge carrier. The field can be defined using the Lorentz force law resulting in

$$\vec{F} = q\vec{v} \times \vec{B}, \quad (2.4)$$

$q$  represents the charge of an electron ( $q = 1.6 \cdot 10^{-19} \text{ As}$ ) and  $\vec{v}$  is the velocity of the charge carrier. The SI unit of the magnetic field is  $T$  (Tesla), which has various SI unit

expressions, i.e.  $\frac{V \cdot s}{m^s}$  or  $\frac{N}{A \cdot m}$  [40].

There are several magnetic field measurement and technology approaches for MEMS-based magnetometers, i.e. Hall effect, fluxgate magnetometer and anisotropic or giant magnetoresistive sensors.

Which approach is used depends particularly on the measurement range and magnitude of the field that is to be measured. For most off-the-shelf consumer-grade magnetometers, MEMS-based Hall effect sensors are used [43]. The Hall effect describes the occurrence of a voltage difference that arises perpendicular to a current carrying electrical conductor inside a magnetic field. This voltage difference, called Hall voltage, is due to the Lorentz force that acts on the charge carrier travelling through the conductor causing an accumulation of charges on one side of the conductor, compare figure 2.3. This creates an electrical field, denoted by  $\vec{E}$ , which results in a force  $\vec{F}_e = q\vec{E}$  on the charge carrier that neutralizes the Lorentz force, such that

$$q(\vec{E} - \vec{v} \times \vec{B}) = 0. \quad (2.5)$$

If the velocity of the charge carrier and magnetic field are perpendicular, the resulting force is perpendicular to both, the magnetic field and current, reducing complexity. The probe can be simplified to a plate capacitor, resulting in the following equation to get the Hall voltage

$$U_H = \vec{E} \cdot b, \quad (2.6)$$

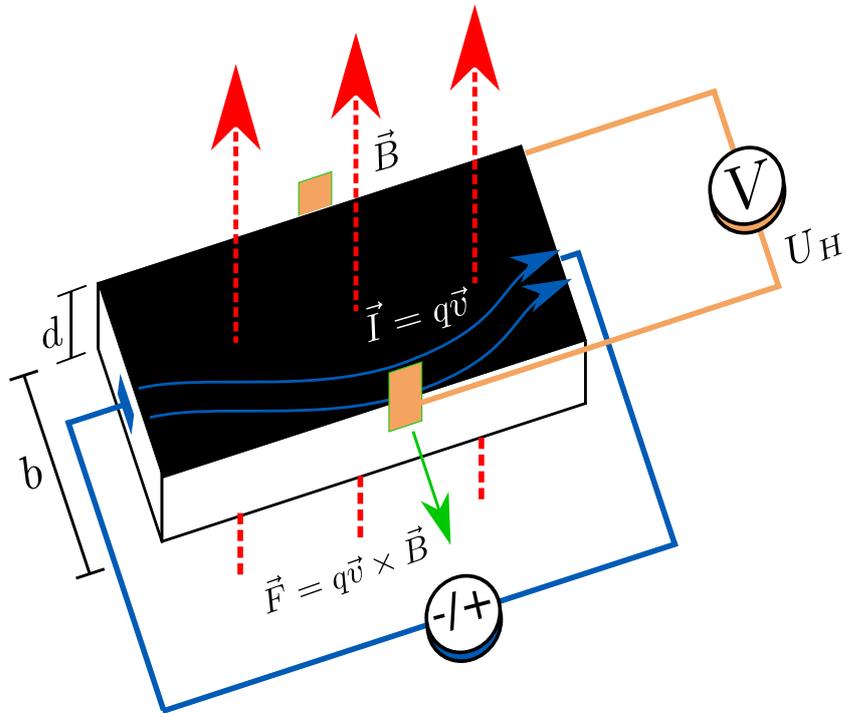


Figure 2.3: Schematic of the Hall effect.

where  $b$  is the distance between the two capacitor plates. Plugging this into equation (2.5) results in

$$U_H = b \cdot \vec{v} \cdot \vec{B}. \quad (2.7)$$

The velocity of the charged particles is given by the current  $I$

$$I = n \cdot v \cdot q \cdot b \cdot d. \quad (2.8)$$

Combining eq. and eq. results in the Hall voltage

$$U_H = A_H \frac{I \cdot \vec{B}}{d}, \quad (2.9)$$

where  $A_H$  is the Hall constant which is equal to  $\frac{1}{nq}$  [40].

Combining three Hall effect sensors, one for each spatial dimension, enables the measurement of the direction of the surrounding magnetic field. For navigation purposes, a three-axis magnetometer is used to measure the direction of the geomagnetic field, which is used as an absolute reference for the heading direction inside a MARG sensor [41]. In an ideal world, a three axis magnetometer would measure the direction of the surrounding magnetic field at any orientation of the sensor, creating an absolute reference for determining true north. If a magnetometer sensor is rotated around all principal axes in a homogeneous magnetic field, recording and plotting the measurement vectors as a point cloud in three-dimensional space, the resulting cloud would be a perfect sphere centered at the origin with a radius equal to the strength of the surrounding field. Every orientation change of the magnetometer would be represented by a unique direction vector (or point) in that sphere [44]. Unfortunately, magnetometers are likewise to all other sensors subject to noise and disturbances, especially extrinsic disturbances. Some extrinsic disturbances or environmental conditions will render a use of a magnetometer to measure a surrounding field completely useless. Section 2.1.2 gives a more detailed overview about MEMS magnetometer noise sources.

### 2.1.2 Noise sources

Noise is an unwanted signal or disturbance that interferes with any measurement signal. For MEMS-based sensors, e.g. accelerometers, gyroscopes, and magnetometers, there are in general two different origins of noise: internal (intrinsic) and external (extrinsic) noise sources. Magnitude and influence of extrinsic noise is tightly coupled to the local conditions, use case and system setup in general, e.g. presence and influence of magnetic or electromagnetic fields, vibrations and system integration (packaging, electro-mechanical integration, temperature distribution and so on). Intrinsic noise sources on the other hand are the limiting factors of MEMS sensors in general and are tightly coupled to their intrinsic design. The intrinsic noise sources are to a certain ex-

tent addressed by the MEMS manufacturer, e.g. through temperature calibration, or at least specified in a given set of environmental conditions. The most critical noise sources for this work originate from the field of application: the human-robot collaboration and use cases with infrastructureless head-worn interfaces, especially regarding the magnetometer measurements. More details follow in the upcoming sections.

The noise sources build a fundamental challenge in MEMS-based navigation and need to be addressed and dealt with in any orientation estimation pipeline.

### Accelerometer noise

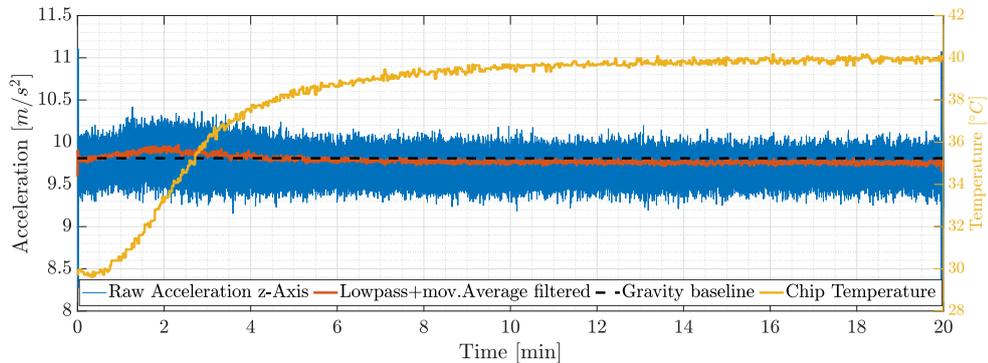
Any measurement will experience noise. To describe the influence of the noise sources on the acceleration measurements science has developed generic inertial sensor error models. This generic model describes the main error sources and their relation on the true acceleration value  ${}^B\vec{f}$ . It is given as follows,

$${}^B\tilde{\vec{f}} = M_{acc} \cdot {}^B\vec{f} + {}^B\vec{b}_{acc} + {}^B\vec{n}_{acc}, \quad (2.10)$$

where  $M_{acc}$  denotes the accelerometers misalignment matrix,  ${}^B\vec{b}_{acc}$  denotes the summation of bias (zero offset) and  ${}^B\vec{n}_{acc}$  is the inherent sensor measurement noise [41].

A misalignment matrix  $M$  is used to align the axis of sensors that measure spatial quantities, e.g. accelerometer, gyroscope or magnetometer triads. It accounts for the non-orthogonality of the sensitive axis and furthermore possible scale factors of each single axis and is usually measured, calibrated, and compensated after the fabrication by the MEMS manufacturer.

The sensor bias  ${}^B\vec{b}_{acc}$  denotes the summation of zero-point errors of a sensor. Literature defines the bias as a near-constant offset that is inherently added onto the measured quantity. In practice, however, this offset consists of a constant component that might vary from switch-on to switch-on but remains constant during the on-time, and a second



**Figure 2.4:** Bias fluctuations of single-axis accelerometer data with respect to time and temperature. The mean signal tends to drift from the initial zero baseline with temperature.

part that varies with respect to time and mostly temperature. The latter is known as bias instability or in-run bias stability and is an essential bottleneck for velocity and position estimation based on double integration of the acceleration. Looking at the frequency domain of the bias instability, it can best be approximated by  $1/f$  pink noise. Figure 2.4 depicts the in-run bias stability for a single-axis MEMS accelerometer facing gravity over a 20-minute period. While the sensor is warming up, the bias (offset from  $1g$ ) varies upwards and downwards. This offset leads to a significant error if a velocity estimation or position estimation is computed from double integration. A low in-run bias stability is a must for any mature inertial velocity or position estimation system.

Besides the bias, every sensor is subject to inherent measurement noise, typically denoted as  ${}^B\vec{n}_{acc}$ . The most dominant effect on this internal noise source for MEMS-based accelerometers is thermal noise. This is due to the inevitable temperature-induced fluctuations of any matter [40]. The seismic mass will extort thermally induced movement that is tightly coupled to the mechanical properties of the MEMS accelerometer. Therefore, a large seismic mass and high damping factor of the resonator will lead to a small intrinsic noise [40]. The noise is assumed as a zero-mean Gaussian noise and quantified using the sensor's spectral density or its square root. In the inertial navigation community, this noise equivalent acceleration is termed as velocity random walk (VRW) [45]. As can be guessed from its name, this inherent noise will have a major influence on the velocity estimation. This is best described if we compare the integration of the random walk noise with a coin flip. At every measurement, the noise might add up or decrease the integrated signal (hence the name random). In the worst possible case, the error might only consist of added noise (extremely unlikely, but possible) and therefore deviate the integrated signal, accumulating and getting worse over time. Even though the noise is assumed to be zero-mean noise, the standard deviation of the noise will eventually lead to drift. This error cannot be compensated without additional references when estimating velocity or position over time with pure MEMS accelerometer measurements but does not play a big role in estimating orientation from gravitational acceleration, since the quantity is not integrated.

The largest extrinsic noise source for MEMS accelerometers is vibration in the system the sensor is mounted on. Quadrotors, for example, are usually equipped with a vibration damping gel between the accelerometer and the quadrotor. In the context of human-robot collaboration, most extrinsic noise is neglectable, since it usually exists at higher frequencies and because the MARG sensor is worn on the head of the human being, acting as a damping between external vibrations and the MARG sensor.

### **Gyroscope noise**

Likewise to the accelerometer, a MEMS gyroscope experiences internal and external noise sources which influence the accuracy of the measurement and introduce noise of different kinds. Since the angular rate measured by the gyroscope is numerically inte-

grated to acquire an orientation estimate, the noise plays a vital role for the anticipated precision of the estimate. Hence this subsection gives a more detailed inside into inertial MEMS noise, even though the noise sources are the same as for the accelerometer that has been described in the prior subsection.

Since MEMS gyroscopes are basically a MEMS accelerometer featuring an oscillating drive, the generic gyroscope noise model is essentially equal to the accelerometer noise model. The measured angular rate  ${}^B\tilde{\omega}$  can be written as the sum of the true angular rate  ${}^B\omega$  and two additional noise sources, leading to [41]:

$${}^B\tilde{\omega} = M_{gyro} \cdot {}^B\omega + \vec{b}_\omega + \vec{n}_\omega, \quad (2.11)$$

where  $M_{gyro}$  is the misalignment matrix of the sensitive gyroscope axis,  $\vec{b}_\omega$  is a term representing gyroscope bias (or offset) and  $\vec{n}_\omega$  the inherent sensor noise.

**Misalignment:** The misalignment matrix is usually measured and calibrated after production at the manufacturing facility and will compensate for misalignment between the different axes to ensure orthogonal measurements and correct scale. If the misalignment is not correctly handled or calibrated, it will introduce an orientation drift proportional to the angular rate of the motion it undergoes [45].

**Inherent noise:** When mathematically describing noise, it is of crucial interest to know the physical noise sources to estimate and describe their effects on the measurement system. The inherent noise of a gyroscope  $\vec{n}_\omega$  is mainly based on thermal noise likewise to the accelerometer and scales with respect to damping, proof mass, driving motion and driving frequency. Since every mass is subject to thermal movement it does make sense to first look at the thermal noise formula for a resonating mass of an accelerometer and apply this to the gyroscope noise considerations. The complete derivation can be found in [40], but the basic idea is addressed shortly. The idea is to derive the noise in terms of acceleration that describes the power spectral density (energy over time) stemming from the thermal movement of a seismic mass, expressing it as a bandwidth independent noise equivalent. The starting point is the velocity, introduced from the thermal movement of the mass to proceed to the power density of the noise. The noise spectral density of a MEMS acceleration due to thermal noise can be expressed as follows [40]:

$$\frac{a_N}{\sqrt{\Delta f}} = 2\sqrt{k_B T \frac{\omega_0}{mQ}}, \quad (2.12)$$

where  $k_B$  is the Boltzmann constant,  $T$  the temperature,  $\omega_0$  denotes the resonance frequency,  $m$  the seismic mass and  $Q$  the quality factor respectively.

The noise equivalent angular rate from thermal noise can be calculated analogously by

equating it with the Coriolis acceleration, resulting in the following [40]:

$$ARW = \frac{\Omega_N}{\sqrt{\Delta f}} = \sqrt{\frac{k_B T}{\omega_0 x_0^2 m Q}}, \quad (2.13)$$

where  $x_0$  denotes the driving frame amplitude [40]. In the navigation genre, this term is also known as Angle Random Walk (ARW) ( $ARW = \frac{\Omega_N}{\Delta f}$ ), describing the variance of the sensor inherent noise  $\vec{n}_\omega$  or  $\vec{n}_{acc}$ , that is assumed to be a zero mean Gaussian white noise ( $\mu_k$ ) with standard deviation of  $\sigma_\mu = 1$  [41], leading to the following algebraic expression

$$\vec{n}_{\omega,k} = \frac{ARW}{\sqrt{\Delta t}} \cdot \mu_k. \quad (2.14)$$

The angle increment of the noise term  $\vec{n}_{\omega,k}$  is expressed as

$$\Delta\Theta_k = \vec{n}_{\omega,k} * \Delta t = ARW \cdot \sqrt{\Delta t} \cdot \mu_k. \quad (2.15)$$

and its variance having the following form

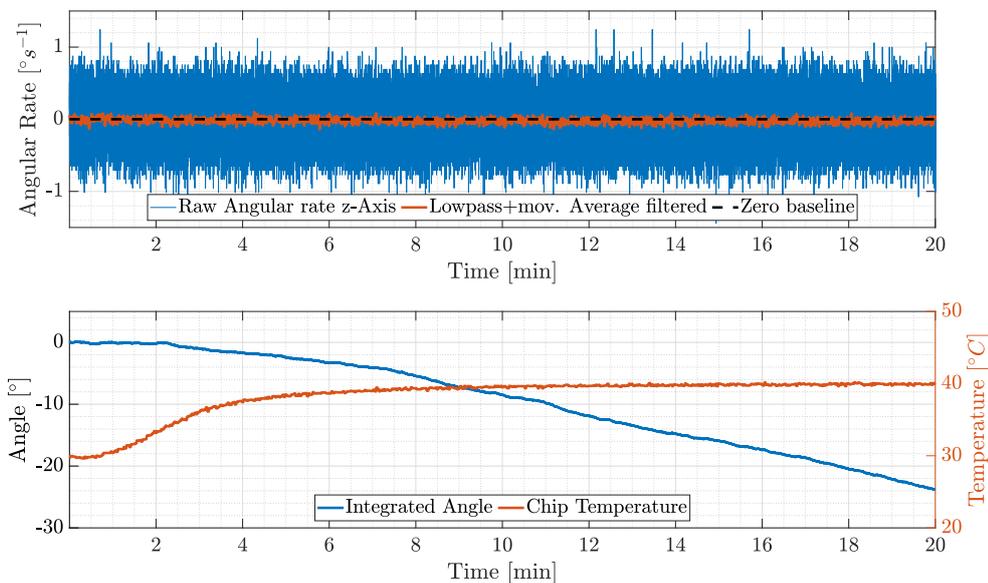
$$\sigma_{\Delta\Theta}^2 = ARW^2 \cdot \Delta t, \quad (2.16)$$

keeping in mind that  $\sigma_\mu$  is 1 and therefore does not show up in this term. If multiple angle increments ( $m$ ) are considered, the variance is given by  $\sigma_\Theta^2 = m \cdot \sigma_{\Delta\Theta}^2$ . From this, it can be seen, that the standard deviation of the angle increments due to the inherent noise can be calculated through

$$\sigma_\Theta = \sqrt{m \cdot \Delta t} \cdot ARW \quad (2.17)$$

[41]. Having the ARW enables to make a guess of the noise distribution for various integral limits, e.g. an ARW of  $1.0 \frac{\circ}{\sqrt{s}}$  means, that the standard deviation of the angle distribution of the inherent noise will be around  $31.6^\circ$  after 1000 seconds ( $t = m * \delta t$ ). The ARW describes random, uncorrelated noise in the measurement steps of a gyroscope. In theory, the average of the integration of the signal should be around zero. In practice, however, when integrating the angular rate measurements, the noise will rise and lower the integration and eventually lead to a drift in the signal. Using the ARW, it is possible to make a statement about the distribution (standard deviation) of the numerical integration of the angles.

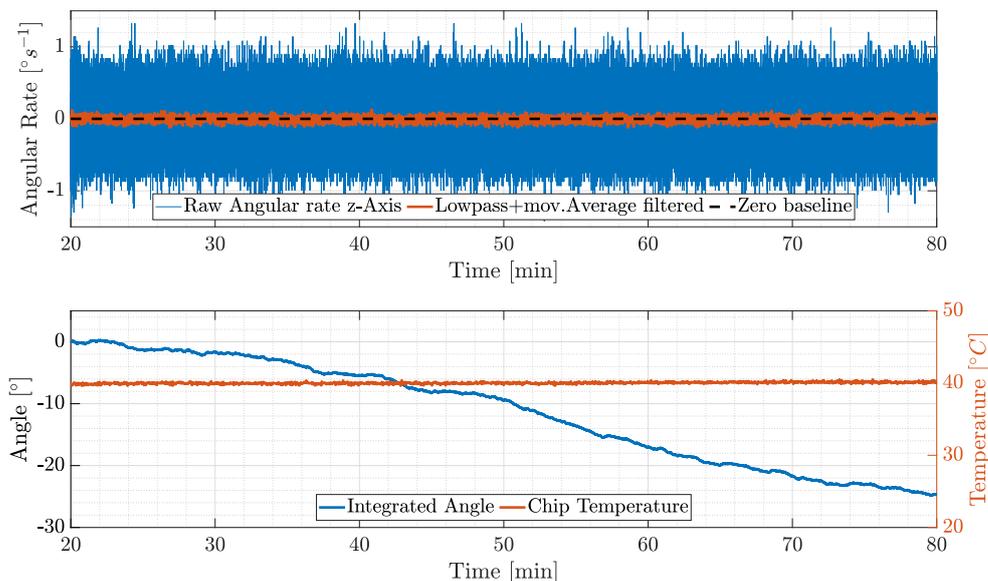
**Bias:** Likewise to the accelerometer, the gyroscope is subject to a DC bias noise term  $\vec{b}_\omega$  that interferes with the numerical integration, since it poses as an offset on the measured quantity. Similarly to the accelerometer, the bias can be separated into a stationary and smaller varying part, compare section 2.1.2, subsection accelerometer noise. Even



**Figure 2.5:** Angular rate depiction of varying bias with changing temperature and its effect on simple angular rate integration. The mean signal tends to drift from the initial zero baseline.

if the device is at rest, the bias will lead to a deviation in the numerical integration of the angular rate with respect to time, if not compensated somehow. For a tactical grade MEMS gyroscope, the bias can be as small as  $0.01^{\circ}s^{-1}$ . Even though this is a small quantity, it will compromise the determination of the gyroscope measurement with every time step by adding this minuscule error with every integration step. After just 10 minutes, this will lead to a  $6^{\circ}$  deviation of the actual orientation, albeit the device did not experience any motion. This constant bias is usually measured over a long period of time under fixed conditions and subtracted afterwards. Unfortunately, the bias drifts over time.

This variation is known as bias instability or in-run bias stability and is usually described as the  $1/f$  noise spectrum of the gyroscope output [46]. The bias error originates from a variety of possibilities, e.g. temperature or g-sensitive bias (a function of the acting acceleration), manufacturing imperfections and so on [41]. Figure 2.5 depicts a gyroscope signal measurement over 20 minutes. The initial bias is sampled over a 20 s time window and subtracted from that point forward. The raw signal is captured at 100 Hz and depicted in blue. To emphasize and visualize the gyroscope bias and reducing the inherent noise  $\vec{n}_{\omega}$ , the signal is low-pass and afterwards moving average filtered and is overlaid in orange. After 20 minutes, the mean signal averages around  $-0.045^{\circ}s^{-1}$ . Integrating this signal results in inevitable angle drift of around  $23^{\circ}$  after 20 minutes, compare figure 2.5. As has been stated, the DC bias changes with respect to temperature. But even if the temperature is held constant using a climate cabinet with closed



**Figure 2.6:** Angular rate depiction of the smaller varying bias at near constant temperature and its effect on angle estimation from angular rate integration. The mean signal tends to drift from the initial zero baseline over a longer period of time.

loop temperature regulation, the bias remains to vary, smaller but steady. This can be seen in Figure 2.6, where the gyroscope signal was measured at constant temperature over an hour. If the sensor is warmed up to  $40^{\circ}\text{C}$  for a period of 20 minutes and the bias is measured and subtracted afterwards, even though the temperature is held constant, the angular rate tends to drift from the zero-bias baseline. Over the course of 60 minutes, the angle results in an accumulation of drift of around  $25^{\circ}$ .

**Noise estimation using Allan variance:** The above described noise terms are typically quantified using measurement data. In the navigation genre the Allan variance method has gained popularity for analyzing the stochastic inertial noise processes. It was originally designed for the statistical analysis of precision oscillators. The Allan variance ( $\sigma^2(\tau)$ ) describes the statistical variance of a signal as a function of the averaging time (or clusters) and enables the ARW, bias instability and the so called Rate Random Walk (RRW) noise identification for gyroscopes as well as accelerometers. The IEEE Standard 647-2006 describes the RRW as "a random process of uncertain origin, possibly a limiting case of an exponentially correlated noise", that is characterized by spectral density of  $1/f^2$  [46]. Both "mechanical gyro as well as rate biased laser gyros exhibit this noise" and as such needs to be measured and considered [46]. For MEMS

gyroscopes, the rate Allan variance ( $\sigma_{\Omega}^2(\tau)$ ) is calculated as follows [46]:

$$\sigma_{\Omega}^2(\tau) = \frac{1}{2\tau^2(n-2m)} \sum_{k=1}^{n-2m} (\theta_{k+2m} - 2\theta_{k+m} + \theta_k)^2, \quad (2.18)$$

where  $n$  is the discrete number of samples,  $\theta_k$  being the numerically integrated angle from angular rate measurements while  $\tau$  is the cluster duration given by  $\tau = m * \tau_0$ ,  $\tau_0$  being the gyroscope sample time and  $m$  denotes the number of samples per cluster. The relation between the Allan variance and Power Spectral Density (PSD) to describe noise is given by the following [46]

$$\sigma_{\Omega}^2(\tau) = 4 \int_0^{\infty} S_{\Omega}(f) \frac{\sin^4(\pi f \tau)}{(\pi f \tau)^2} df, \quad (2.19)$$

where  $S_{\Omega}(f)$  is the power spectral density of a stationary random process. From this relation, one can derive the noise coefficients from the Allan deviation plot ( $\sigma(\tau)$ ) by equating the Allan variance with the PSD of the noise term. For the ARW, the PSD (variance of the white noise) is given by  $S_{\Omega}(f) = ARW^2$ , substituting it into (2.19) yields

$$\sigma^2(\tau) = \frac{ARW^2}{\tau}. \quad (2.20)$$

This equation represents a line of slope  $-1/2$  when plotted on a log-log plot of the Allan deviation versus  $\tau$ , such that  $ARW$  can be read from the line at  $\tau = 1$ . The procedure is the same for the bias instability estimation. The power spectral density is given by [46]

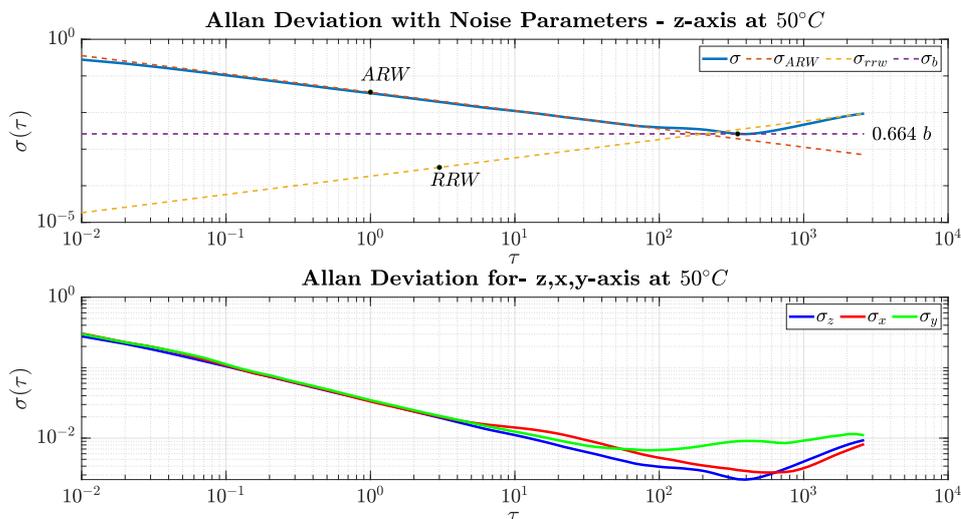
$$S_{\Omega}(f) = \begin{cases} (\frac{b^2}{2\pi}) \frac{1}{f} & f \leq f_0 \\ 0 & f > f_0 \end{cases}, \quad (2.21)$$

where  $b$  is the bias instability coefficient and  $f_0$  is the cut-off frequency. Substitution and integration yield to an equation that simplifies if  $\tau$  gets much longer than the inverse of the cut-off frequency, resulting in [46]

$$\sigma^2(\tau) = \frac{2b^2}{\pi} \ln 2. \quad (2.22)$$

Likewise to the  $ARW$ , the value for  $b$  can be read of this line, which is a line with a slope of zero on the Allan deviation log-log plot. The value scales with  $\sqrt{\frac{2 \ln 2}{\pi}} = 0.664$ . Finally, the rate random walk is characterized by [46]

$$S_{\Omega}(f) = (\frac{RRW}{2\pi})^2 \frac{1}{f^2}. \quad (2.23)$$



**Figure 2.7:** Allan Deviation plot of a tri-axis gyroscope of ICM20948 at  $50^{\circ}C$ . The above plot indicates the slopes of the Allan deviation to identify parameters of noise for the z-axis of the MEMS gyroscope. The below plot shows the Allan deviation curves for all three axes.

Again substituting and integration yields the Allan variance relation

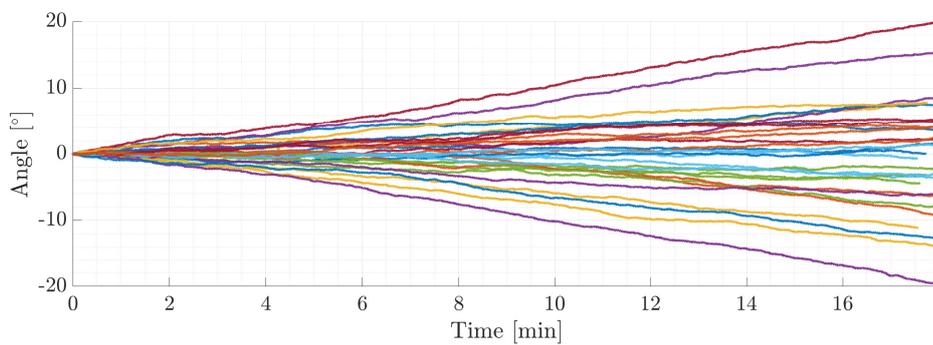
$$\sigma^2(\tau) = \frac{RRW^2\tau}{3}. \quad (2.24)$$

This equation presents a line with a  $1/2$  slope on the Allan deviation plot at  $\tau = 3$ . Using measurement data, it is possible to estimate the characteristic noise sources of MEMS gyroscopes directly from the Allan deviation plot. For the sensor system used here (ICM-20948 [43]), the Allan deviation plot is given as follows. The measurement was conducted using a stationary gyroscope in a climate cabinet at a constant temperature of  $50^{\circ}C$  (to prevent any change in temperature at a lower level), for a total time of 180 minutes. The first 20 minutes are not considered in the measurement data to ensure that the gyroscope and climate cabinet have reached a constant temperature level. It is important to notice, that these measurements have been made under stationary conditions within controlled temperature regions. Temperature fluctuations will inevitably lead to a change in the noise parameters. The noise parameters are identified as follows: The formulation and relation as PSD of the noise enables a direct comparison between different gyroscopes and help to estimate and reduce the effect of the noise on the angular rate measurements.

However, especially regarding the bias noise sources and varying coefficients for the different gyroscopes axis (compare figure 2.7), one can conclude, that numerical integration of angular rate measurements will experience drift that can hardly be reduced or

**Table 2.1:** Noise parameter estimation from Allan variance over constant temperature of  $50^{\circ}\text{C}$  for the ICM-20948 gyroscope.

	ARW	RRW	b
x - axis	$0.0348 \frac{\circ\text{s}^{-1}}{\sqrt{\text{Hz}}}$	$0.00025 \frac{\circ\text{s}^{-1}}{\sqrt{\text{Hz}}}$	$0.0049 \frac{\circ}{\text{s}}$
y - axis	$0.0345 \frac{\circ\text{s}^{-1}}{\sqrt{\text{Hz}}}$	$0.000519 \frac{\circ\text{s}^{-1}}{\sqrt{\text{Hz}}}$	$0.0137 \frac{\circ}{\text{s}}$
z - axis	$0.0359 \frac{\circ\text{s}^{-1}}{\sqrt{\text{Hz}}}$	$0.000316 \frac{\circ\text{s}^{-1}}{\sqrt{\text{Hz}}}$	$0.0039 \frac{\circ}{\text{s}}$

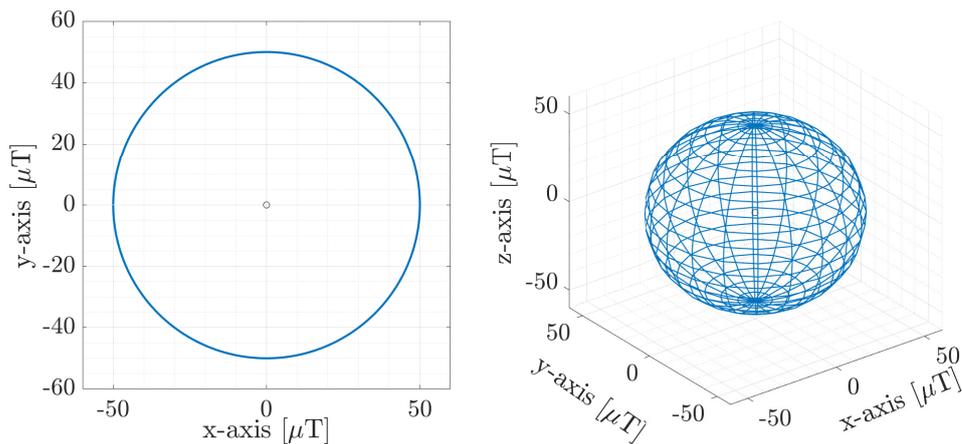
**Figure 2.8:** Gyroscope noise on numerical angle estimation through integration for 30 trials from the same sensor at  $40^{\circ}\text{C}$ .

modelled. Figure 2.8 precisely describes this. The figure depicts 30 trials of stationary gyroscope measurements. The measurements stem from the same sensor in the exact same setup. Prior to every recording, the sensor was warmed up to  $40^{\circ}\text{C}$  for a period of 15 minutes and initial bias at that point was subtracted. The numerical integration process leads to a noisy distribution of the angle estimates spanning between  $0^{\circ} \pm 20$ . Therefore, additional references are needed in order to avoid and compensate error growth when estimating angles over a longer period of time and with varying environmental conditions, e.g. temperature.

## Magnetometer noise

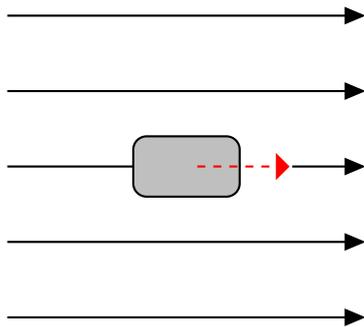
MEMS magnetometers are subject to intrinsic as well as extrinsic noise sources. Since the magnetometer is used to directly measure the geomagnetic field, without the need for any additional mathematical integration, the intrinsic noise does not pose as the major error source. Extrinsic noise, however, directly interferes with the magnitude and direction of the surrounding magnetic field and thus directly affects the measured quantity. The most dominant extrinsic noise sources in the magnetic domain are the so-called hard and soft iron effects. Both effects do substantially disturb the measurement and must be compensated by appropriate methods. To understand the effects and methods to accommodate for the noise we need to take a look at the ideal magnetometer measurements.

On the sensor level we can assume, that the earth's magnetic field is locally homogeneous and constant. A tri-axis MEMS magnetometer can be used to measure the direction and magnitude of this field to enable a compass-like orientation estimation. Turning the sensor around  $360^\circ$  within this field and plotting the measured data points from the respective axes against each other one obtains the ideal magnetic field, which can be represented as a circle (2D) or a sphere (3D). The radius of the sphere is equal to the magnitude of the geomagnetic field, which is around 48 microtesla at our current longitudinal and vertical axis. We have a discrete and unambiguous vector for every orientation change. Hard and soft iron effects deform this ideal sphere which makes the direction measurement ambiguous.

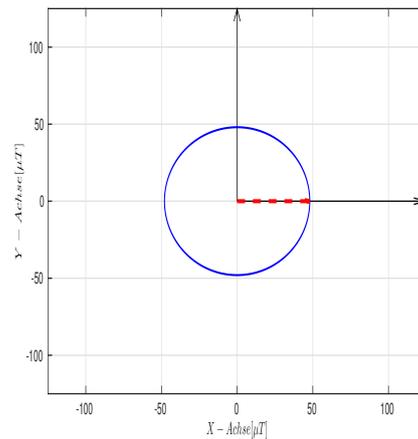


**Figure 2.9:** Measurement values for a calibrated magnetometer in a constant magnetic field. The left figure depicts the measurement values for the x-y plane after a  $360^\circ$  rotation about the z-axis. The right figure depicts the ideal measurement values for a magnetometer around all sensitive axes if the measurement vectors are scattered against each other.

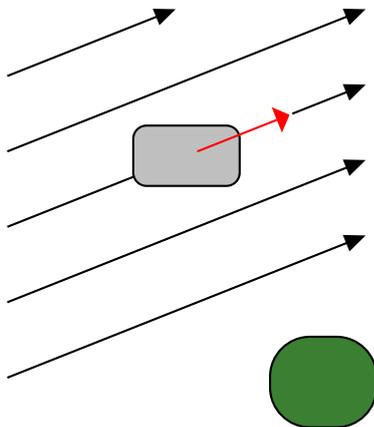
**Hard iron effect** The hard iron effect describes a superimposition of a magnetic field that is to be measured (e.g. earth magnetic field) by an external magnetic field. The overlapping of the fields changes both the direction and magnitude of the surrounding field. This effect is mainly caused from permanent magnetic materials [47]. The extent to which the disturbance affects the magnetometer data can in principle be divided into two cases: a) the permanent magnetic disturbance stays fixed with respect to the global coordinate system or b) the disturbance moves relative to the global frame.



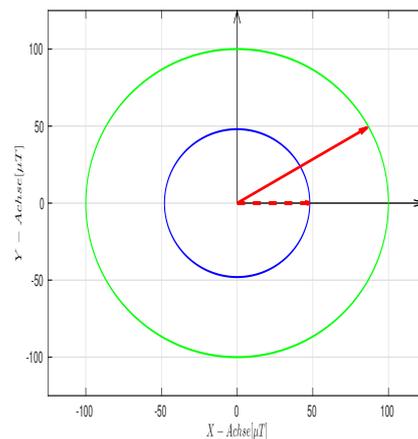
(a) MARG sensor (gray) inside constant magnetic field and the resulting measurement vector (red).



(b) Measurement vector (red) along a 360°-rotation (blue) (one plane).



(c) MARG sensor in constant permanent magnetic field (green), fixed w. r. t. the world coordinate frame.

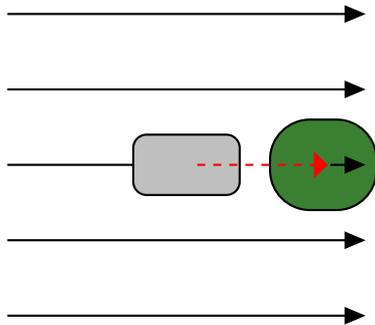


(d) Measurement vector (red) near fixed hard iron distortion along a 360°-rotation.

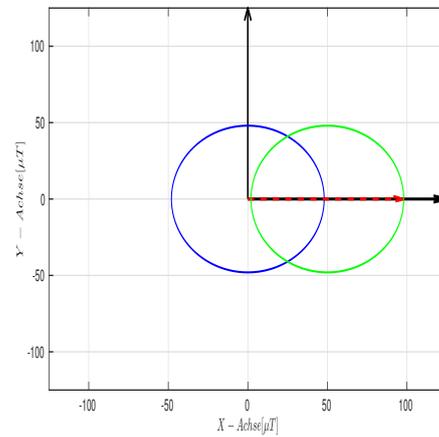
**Figure 2.10:** Illustration of a fixed permanent magnetic influence (hard iron effect) on the magnetic field measurement vector.

**a) The permanent magnetic disturbance stays fixed.** If there is a permanent magnetic source within the vicinity of the magnetometer that is fixed at a constant position and orientation with respect to the earth's magnetic field, the direction and magnitude of the magnetic field lines have an additive effect on the earth's magnetic field. The resulting magnetic field lines act homogeneously on the sensor at the sensor location. The additive component will at all times act on every sensitive axis, resulting in a constant magnetic field scale. A change in orientation can thus still be determined, but the stationary disturbance compromises the field lines in the direction of the earth's magnetic field and thus leads to a change in the directional vector. This disruption can be counteracted by simple recalibration or scaling.

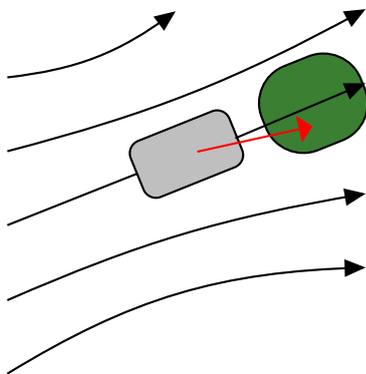
**b) The permanent magnetic disturbance moves inconsistently.** If a permanent magnetic interference lies within the vicinity of a magnetometer and changes its orientation or position with respect to the earth's magnetic field, the magnitude and direction of the magnetic field lines change depending on the relative pose between the magnetometer and the magnetic material. The resulting field lines act homogeneously on the sensor at the sensor location, but do not have a constant direction. The measured amount and direction vector cannot be predicted, unlike from the non-disturbed field lines (see figure 2.10), the direction is ambiguous. This disturbance cannot be compensated by trivial methods. A special case of this disturbance arises if the magnetic source moves along with respect to the magnetometer. For example, if a permanent magnetic source is aligned with a magnetometer axis and moves conjointly, the permanent magnetic interference acts as a constant offset with respect to the sensitive axis that points towards the interference (see figure 2.13). This effect can be compensated by an offset subtraction (averaging after rotation in each axis).



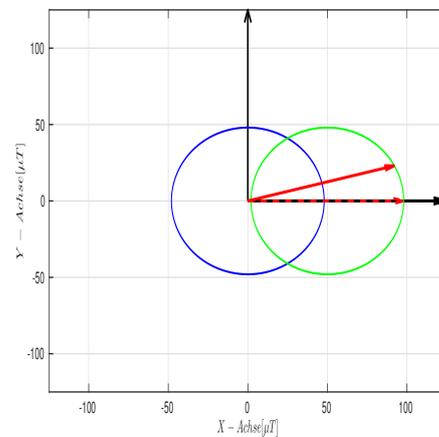
(a) MARG sensor in homogeneous magnetic field and the resulting measurement vector (red).



(b) Measurement Vector along a 360° rotation, without disturbance (blue) and with disturbance (green).



(c) MARG sensor (gray) with constantly rotating disturbance (green), inconsistent to earth magnetic field and resulting measurement vector

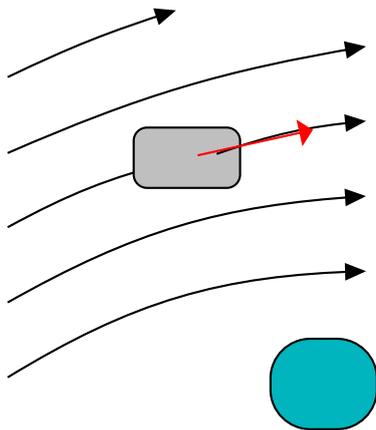


(d) Measurement vector (red) along a 360° rotation, without disturbance (blue) and with disturbance (green).

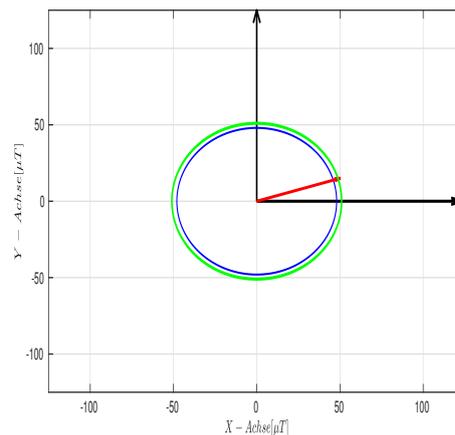
**Figure 2.11:** Depiction of hard iron effect of a permanent magnetic source, inconsistent with respect to earth magnetic field.

**Soft iron effect** The soft iron effect describes the distortion of magnetic field lines around any ferromagnetic material. The influence of the distorted field lines on the measurement is subject to the distance between the ferromagnetic material and the sensor. Distorted field lines lead to a deflection of the measured magnetic field vector that scales with respect to distance and shape of the object. The soft iron effect deviates/deforms the measurement vector along the sensitive lines, which results in an ellipsoid-like shape for a full magnetic field measurement plot. The eccentricity of the ellipsoid changes with respect to the direction of the disturbance [47]. The measurement vector will result in a wrong directional vector measurement. Likewise to the hard iron effect, a ferromagnetic interference in the vicinity of a magnetometer will have two different effects on the measurement of the field.

**a) The ferromagnetic disturbance stays fixed.** If the ferromagnetic interference stays at a constant pose with respect to the earth’s magnetic field (no change in orientation or position), the direction of the field lines will be bent. The resulting field lines continue to act homogeneously on the sensor at the sensor location (due to the small size). A change in orientation can thus still be determined, but the stationary disturbance compromises the field lines in the direction of the earth’s magnetic field and thus leads to a change in the directional vector. Likewise to the constant hard iron effect, this disturbance can also be counteracted by simple recalibration or scaling.



(a) MARG sensor (gray) and soft iron disturbance (blue), fixed with respect to earth magnetic field and the resulting measurement vector (red).

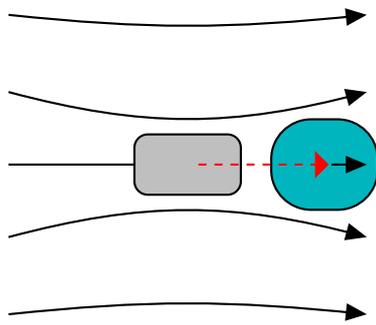


(b) Measurement vector (red) along a 360° rotation, without disturbance (blue) and with disturbance (green).

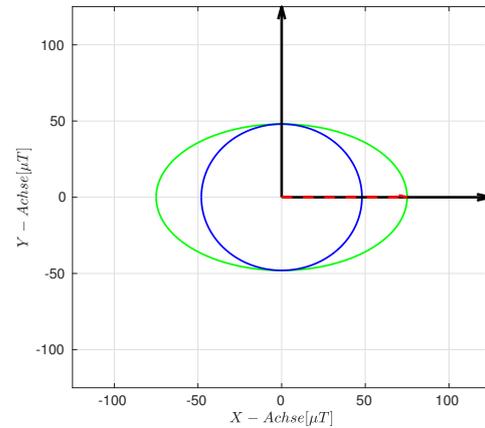
**Figure 2.12:** Soft iron effect of a ferromagnetic disturbance fixed with respect to the earth’s magnetic field.

**b) The ferromagnetic disturbance moves inconsistently.** If a ferromagnetic disturbance changes its orientation or position with respect to the earth's magnetic field, the direction of the field lines and the magnitude of the field change. The resulting field lines act homogeneously on the sensor at the sensor location, but do not have a homogeneous direction. The magnitude and direction are ambiguous and not predictable. A unique change in orientation cannot be determined. Likewise to a relative moving permanent magnetic source, it is a non-trivial task to eliminate this effect.

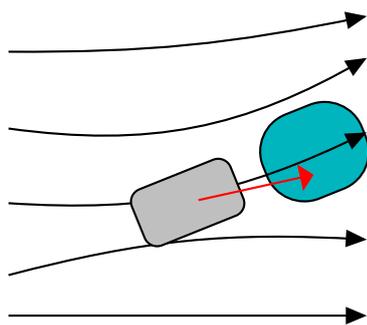
A special case arises if the disturbance moves along with respect to the magnetometer.



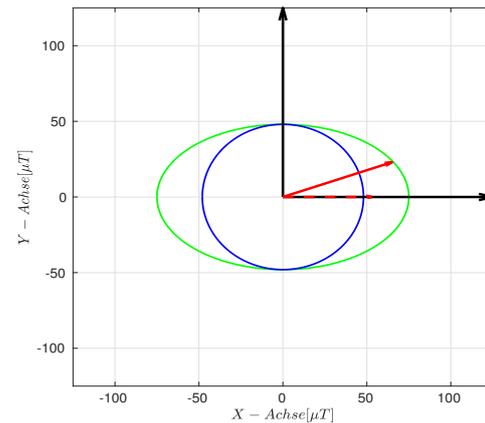
(a) MARG sensor (gray) in distorted magnetic field and resulting measurement vector (red).



(b) Measurement vector (red) along a 360° rotation, without disturbance (blue) and with disturbance (green).



(c) MARG sensor (gray) with constantly rotated disturbance (blue), inconsistent to earth magnetic field and resulting measurement vector (red).



(d) Measurement vector (red) along a 360° rotation, without disturbance (blue) and with disturbance (green).

**Figure 2.13:** Depiction of soft iron effect of a ferromagnetic source, inconsistent with respect to the earth's magnetic field.

If a ferromagnetic source moves conjointly to one magnetometer axes, the field lines are bent. The magnetic field vector gets a vectorial part in another axis. If we plot the axis values after a full rotation against each other, the magnetometer data forms an ellipse (in 2D) and an ellipsoid in 3D.

Prior to using magnetometer data in any orientation estimation pipeline it needs to be calibrated to the surrounding magnetic field. This includes all stationary and non moving ferro- and permanent-magnetic materials (sources) that are present on the sensor system or carrier board, for example metal PCB traces and inductors (DC-DC switch) near the sensor. All other non stationary or possible varying interference should be absent while calibrating the sensor to acquire the field that is to be measured (earth magnetic field). Th calibration is usually achieved using an ellipsoid to sphere mapping procedure. The magnetometer is rotated around every axis while the measurement vectors are sampled. The data usually resembles an ellipsoid (soft iron effect) that is displaced in three dimensions from the origin (hard iron effect). Since we strive to get a unit sphere to have unambiguous measurement vectors for every direction, a common and straightforward method is to find the parameters of an ellipsoid using linear regression that would best fit the accumulated data and compensate the ellipsoid backwards to a sphere. This is usually solved using Least Square [48]. An off-center rotated ellipsoid represented in terms of linear regression is given by

$$ax^2 + by^2 + cz^2 + 2dxy + 2eyz + 2fzx + 2gx + 2hy + 2iz = 1, \quad (2.25)$$

where  $x, y, z$  represent the three axis magnetometer sample and  $[a, b, c, d, e, f, g, h, i]$  the 9 unknown parameters that are to be solved for [49]. Rewriting the equation in matrix form to enable a least square solution to be found results in

$$N = D \cdot \vec{v}, \quad (2.26)$$

with  $N, D$  and  $\vec{v}$  are given as follows

$$D = \begin{pmatrix} x^2 & y^2 & z^2 & 2xy & 2xz & 2yz & 2x & 2y & 2z \end{pmatrix}_{n \times 9}, \quad (2.27)$$

$$\vec{v} = \begin{pmatrix} a & b & c & d & e & f & g & h & i \end{pmatrix}_{9 \times 1}^T, \quad (2.28)$$

and

$$N = \begin{pmatrix} 1 \\ 1 \\ \dots \end{pmatrix}_{n \times 1}, \quad (2.29)$$

where  $n$  is the number of magnetometer samples.

To solve for the above unknowns at least nine different points are needed. The Least Square error approximation for the above given equation is computed using the pseudo-

inverse of the data matrix, resulting in

$$\vec{v} = D^T \vec{v} \cdot D^{-1} D^T \cdot N. \quad (2.30)$$

The unknown parameters are then used to build auxiliary matrices that map the offset, rotation and radii of the ellipsoid equation onto the unit sphere. This process is based on equating (2.32) with an alternative formulation of an arbitrary ellipsoid, that is given as follows [48]

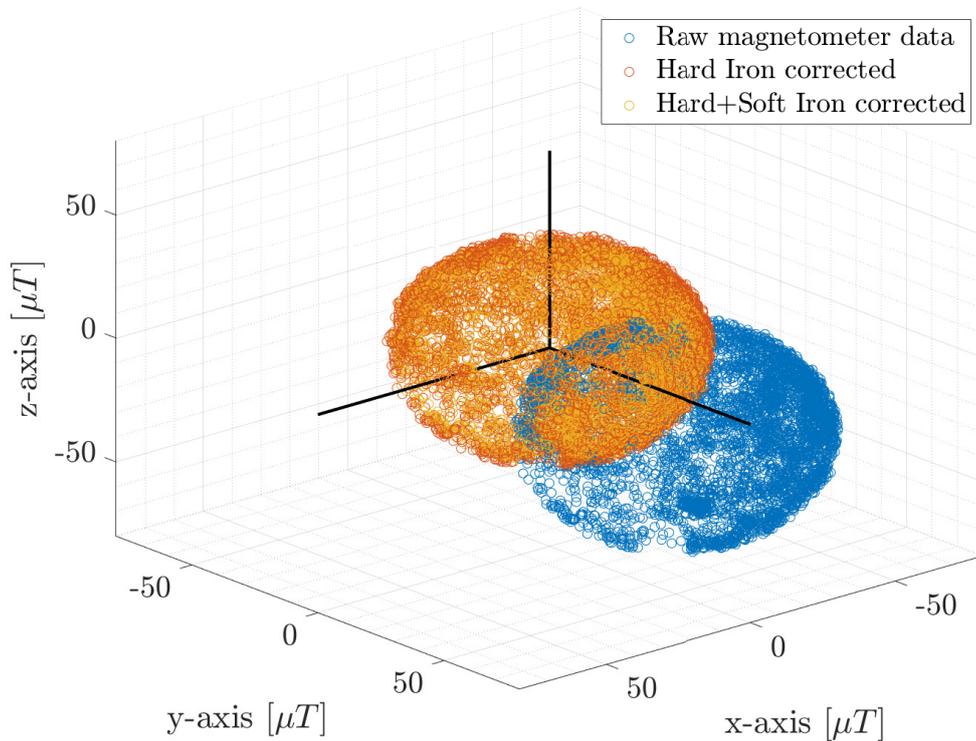
$$1 = (\vec{x} - \vec{b})^T A (\vec{x} - \vec{b}), \quad (2.31)$$

expanding the equation leads to

$$1 = \vec{x}^T A \vec{x} - 2 \vec{x}^T A \vec{b} + \vec{b}^T A \vec{b}, \quad (2.32)$$

where  $\vec{b}$  represents the offset and  $A$  the rotation and scaling respectively.

Equating (2.31) with (2.32) and decomposing the final matrix into its eigenvectors yields to a parameter set describing the rotation and radii of the ellipsoid used to map it to the



**Figure 2.14:** Magnetometer measurements and the hard as well as soft iron corrected data points.

unit sphere. For more details the reader is referred to [48].

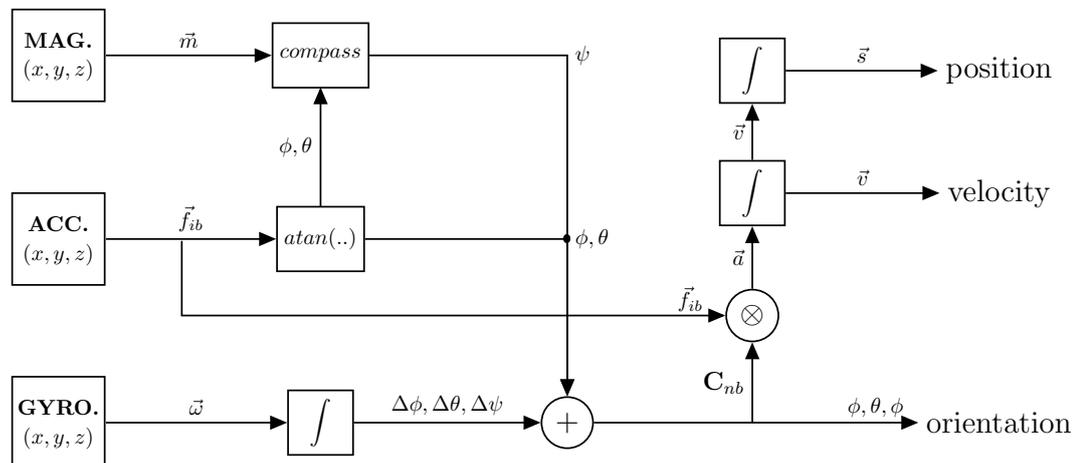
An example for this initial calibration with real measurement is depicted data in figure 2.14. The raw measurement samples (blue) are first translated into the origin using the offset  $\vec{b}$  (orange), also known as hard iron correction, and afterwards scaled and rotated into the unit sphere (yellow), commonly referred to as soft iron correction. If using this technique, there is one crucial condition to consider here: the sampled data points should represent the ellipsoid, meaning it should at least cover a sufficient amount of samples in all planes. This can be measured using gyroscope data to ensure, that the samples collected during rotation resemble an ellipsoid [48].

### 2.1.3 MARG sensor data fusion principles

The combination of a three-axis accelerometer, three-axis gyroscope and three-axis magnetometer is termed MARG sensor and can be used for navigation purposes. The underlying data fusion process is generally defined as a computation rule for determining the location, speed, and position of a navigation system. In case of inertial and magnetic navigation systems (MARG sensors) this data fusion is also termed strapdown inertial navigation integration algorithm. Strapdown refers to a MARG sensor that is directly "strapped down" to an object's body (e.g. vehicle), meaning that the object and sensor axes move conjointly.

Various approaches have been researched and developed to fuse MARG sensor data to calculate orientation, velocity and position respectively, including adaptive Filters, e.g. Kalman Filters (linear, extended, unscented) and constant gain filters, e.g. Madgwick's gradient-descent filter or the Mahony filter [25]. Nevertheless, all data fusion approaches follow a similar procedure which will be called strapdown algorithm from now on and will be described below.

**The strapdown algorithm** generally defines a computation rule for determining the orientation, speed, and position of a navigation system. Gyroscopes enable the calculation of orientation by simply integrating the measured angular rate vector over time.



**Figure 2.15:** Simplified block diagram of MARG sensor-based strapdown algorithm for inertial navigation calculation. A tri-axis magnetometer and accelerometer give an absolute orientation estimate that is used to correct the relative orientation estimation from angular rate integrations from a tri-axis gyroscope. The orientation is afterwards used to calculate the direction of gravity and subtract it from the overall acceleration measurement to double integrate the quantity to get velocity and position respectively.

This does unfortunately only give a relative information, since the gyroscope does only measure rotations about its own axis and not with respect to a world coordinate frame. To enable a full orientation estimation an initial orientation should be supplied. The initial orientation can be calculated based on the direction of the measured gravity vector and the direction of the magnetic north pole which give a three-dimensional reference. From this point forward, one can calculate new orientation based on the integration of the gyroscope measurements. This orientation is used to determine the direction of gravity and subtract it from the accelerometer data, leaving only the amount of acceleration from the motion of the object itself. Integration of this acceleration leads to a velocity estimate. Integration of this velocity estimate leads to a position estimate. Figure 2.15 illustrates a simplified strapdown algorithm.

Under ideal conditions, this "simple" approach would be sufficient to calculate the pose of an object in space with respect to a navigation frame. Unfortunately, all sensors are subject to noise and disturbances and will therefore diverge from this ideal solution. Due to the double integration of the non-gravitational acceleration, the accuracy of the pose determination heavily depends on the quality and precision of the orientation determination. Since gyroscopes suffer from DC bias that changes over time and temperature, the simple integration scheme will accumulate drift and result in growing orientation errors with respect to time. Every minuscule error in the orientation will result in a wrong estimate of the gravity vector and thus result in a false velocity vector. This error is further amplified in the position estimate since the error will be integrated even further. In MARG sensors, this orientation accumulation error is reduced by correcting the gyroscope-based orientation estimation using the global reference orientation calculated based on accelerometer and magnetometer measurements. This is also the case for magnetic disturbances. These deviate the direction and magnitude the measured geomagnetic field vector and will introduce an erroneous reference vector which will ultimately deviate the orientation estimation with respect to the yaw or heading axis. A considerable amount of work has to be done to enable a sufficient pose determination, starting at the bottleneck which is orientation estimation.

## 2.2 Mobile eye tracker

This section will give an overview of existing mobile eye tracking technology. This information is crucial to understand the research efforts made here, since they mostly combine mobile eye tracking cameras (either eye or scene cameras) with MARG sensors. The analysis of the direction or the location of interest of a user through eye tracking is key for many applications in various fields, i.e. human-computer interfaces (gaze mouse), human-machine interfaces, medical diagnostics and many more [50]. Therefore, fast and reliable eye tracking devices and software have been heavily researched [51]. Eye trackers can be separated into stationary or mobile devices. Stationary eye trackers are fixed in position referenced to the world frame. The user is bound to a minimal motion

space for the measurement system to work. An example for a stationary eye tracking device is the Tobii 4C eye tracker, used in [3]. It is a video-based stationary eye tracking device that consists of a camera which observes the user's eyes and maps the tracked gaze to a defined fixed surface (e.g. screen monitor) [3]. Mobile devices on the other hand usually consist of a wearable device that does not depend on a fixed infrastructure in terms of a motion space.

There are various methods to measure the movements of a human eye and determine the gaze direction. When it comes to mobile applications, this variety narrows down to infrastructureless measurement methods that do not include any stationary measurement systems and therefore do not limit the head motion space. Alsharif [20] divides eye tracking into three distinctive groups: scleral search coils, electrooculography, and video-based eye tracking.

**Scleral search coils** are based on a semi-invasive technique to measure eye movement with very high precision. The scleral search coil methods consist of a contact lens with an embedded coil (inductor) inside a homogeneous magnetic field. The magnetic field induces a voltage in the coil, which is used to measure the movement of the eyeball. This method has become the gold standard for eye movement tracking but is rather uncomfortable and does require a sufficient hardware setup when used for mobile tracking [52].

**Electrooculography** is a method based on the biopotentials (dipole) of the human eye. A set of electrodes is distributed across the eye to measure the dipole (direction of eye). This method is especially cheap and fairly simple to use but comes at a drawback in possible resolution and long-term reliability due to the changing impedance of the electrodes. Furthermore, the electrodes need to be placed on specified regions which usually takes a lot of time and an experienced instructor [20].

**Video-based eye tracking** is the process of estimating pupil positions from a near eye camera using computer vision methods. Due to the advances in camera miniaturization and computer vision in the last decade, video-based eye tracking is the most popular and feasible technique. Mobile eye tracking devices usually consist of a frame that is worn like a pair of glasses, a mono- or binocular eye camera fixed to the frame monitoring the pupil and a scene camera to merge calibrated pupil positions to a gaze point in the world frame [51]. The pupil position estimation from video-based eye tracking can be achieved from various computer vision approaches. The most notable are corneal reflection, glint vectors, Purkinje images and bright or dark eye detection [20]. Most modern mobile eye tracking devices either feature a MARG or an IMU sensor or the eye tracker can

be extended by a custom or third-party sensor board. The IMU is used to measure the head rotation and counteract the effect of simultaneous head and eye movements [53]. Besides its non-invasive character and ease of use, video-based eye trackers give more parameters for eye tracking research measurements, including pupil size estimations. In this thesis, a lightweight mobile eye tracker that gained popularity in the research community over the past years due to the open-source software and affordable pricing [54] is used for measuring pupil motion and estimating gaze directions to a) support the IMU or MARG sensor heading estimation accuracy (compare chapter 5) and b) use eye motion and gaze as discrete and continuous control signals for the robot control application.

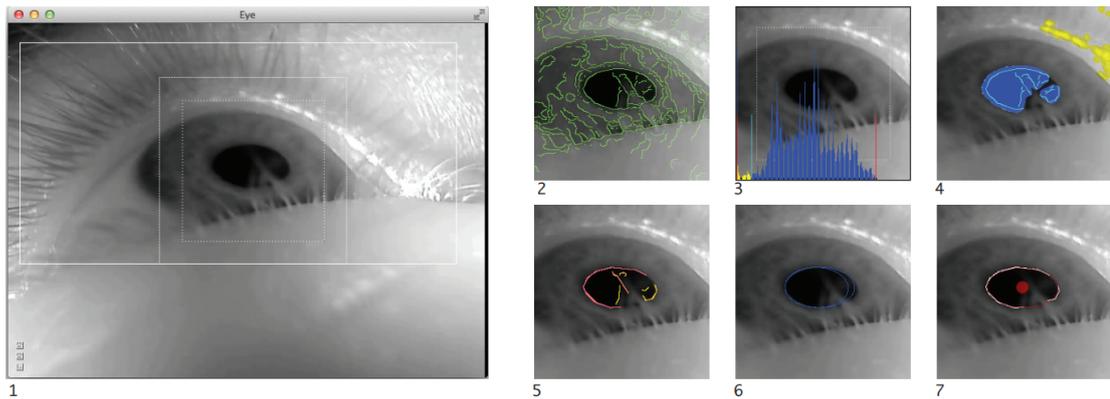
### 2.2.1 Mobile video-based eye tracking

There exist different approaches for mobile video-based eye tracking to estimate gaze and pupil parameters from an eye camera image. They are divided into three groups – regression-, model- and learning-based approaches for eye gaze estimation [20]. Regression- and model-based methods rely on the detection of two-dimensional pupil features from an eye camera image whereas learning-based approaches use a machine learning pipeline to estimate gaze from eye images onto a scene camera image, using the eye images as the input [55].

Learning-based eye tracking approaches employ a machine learning pipeline to estimate gaze from an eye image, typically trained on a massive dataset consisting of eye camera images alongside ground truth gaze positions. In the last year there has been major success implementing these methods in head-worn eye trackers leading to a robust and calibration-free gaze estimation in a commercially available device (e.g. pupil invisible [55]). These types of eye trackers, however, are not as precise as model- or regression-based approaches and are for example currently not capable of detecting accurate fixations [56].

Regression-based methods directly map the two-dimensionally detected pupil features using a low-order polynomial (order 5-7) onto a solution surface, i.e. the scene camera image [51]. Model-based methods on the other hand use the detected two-dimensional features to estimate a geometric three-dimensional model of the eye and employ this model to estimate gaze points [57]. Regression- and model-based eye trackers are the dominating eye tracking technologies on the market. They share the same initial step of pupil parameter detection from the image and are both in need of an initial calibration step to estimate calibration parameters that essentially allow the gaze direction estimation [56].

The feature detection of pupil parameters from eye images is the bottleneck when it comes to video-based eye tracking. The estimation of the pupil center, pupil size and further parameters is depending on the resolution of the eye image and its illumination.



**Figure 2.16:** Pupil Labs eye detection pipeline [51]: "1 - Grayscale eye tracker image, 2 - Canny edge detection, 3 - define dark region as offset from lowest spike in histogram, 4 - Filter Edges, exclude reflections (not in dark, yellow), 5 - Extract contour of edges, 6 - Extract plausible contour candidates using ellipse fitting, 7 - detected ellipse, found through augmented combinatorial search". Image source [51], copyright Pupil Labs.

To enhance contrast in these images most commercial eye trackers use Infrared (IR) LEDs at a wavelength of around 780 – 880  $nm$  near the eye camera. The eye camera is usually equipped with an IR bandpass filter to further reduce ambient lighting effects which significantly enhances contrast in the eye image. Based on the axis alignment of the illumination source and the optical axis of eye camera, the eye tracking approach can be divided into dark and bright pupil tracking. Dark pupil tracking has become the dominant source of eye tracking technology since both the lighting and image source do not be in direct axis alignment. Beside illumination, the IR source introduces pupil corneal reflections, also known as glint. Gaze positions can be estimated based on the pupil-glint vectors, which is the relative distance between the pupil center and the corneal reflection from the IR source [58]. Besides the popularity of glint vector-based gaze estimation in commercially available devices, the Pupil Core binocular headset used in this work does not use the glint vector. Avoiding the glint vector, which is essentially a reflection, widens up the overall usability of the eye tracking glasses, in such a way that it can be used by humans wearing contact lenses and even prescription glasses, since these would interfere with the glint vector [51]. The detection pipeline from Pupil Labs, compare figure 2.16, employs computer vision methods to detect the contour of the pupil and estimate its position in eye image coordinates – this is the 2D detector. The results of the 2D detection are then used in a regression-based approach using a bivariate polynomial to map pupil position onto the scene camera image. Besides the 2D detector, the Pupil Labs real-time framework features a 3D model-based detection that fits a 3D geometric eye model onto the 2D pupil contour without the need of glint vectors. The 3D model is more robust to slippage, but less accurate compared to the 2D model [51].

Since a head mounted video-based eye tracker features a scene camera, it is possible to associate the gaze estimation with an object or area of interest in that scene view. This can be achieved based on pose estimation of the user head with respect to the object and three-dimensional gaze point transformation. Estimating three-dimensional gaze is either done by gaze ray intersection from binocular eye tracking estimates in a 3D eyeball model [59] or more straightforward and simpler by mapping the gaze on a depth image from a stereo or optical Time-Of-Flight (ToF) camera.

### 2.2.2 Video-based eye tracker noise sources

Likewise to every camera sensor, a head-worn video-based eye tracker is subject to internal and external noise sources. The internal noise sources are tightly bound to the underlying technology and structure of the camera sensor, e.g. Complementary Metal-Oxide-Semiconductor (CMOS) spatial noise is significantly larger when compared to a Charge-Coupled-Device (CCD) sensor, due to the single-pixel amplifier versus a full-pixel-row amplifier for a CCD – see [60] for more details. CCD sensors are sensitive to light and can experience blooming, which is the spreading of charge into neighboring pixels once a pixel has reached its full-well capacity. CMOS sensors do not suffer from this effect, since every pixel is isolated and not coupled in series. Furthermore, CCD sensors are slow compared to CMOS sensors when it comes to readout capability, due to the row-based readout circuitry bottleneck. Most of these internal noise sources are either addressed by the manufacturer of the sensor or need to be accounted for depending on the desired application (lightning conditions, motion, optical distortion, etc.). The strength and weaknesses arising from the imager's internal structure define the use cases for the sensor [60].

Modern head-worn eye tracking cameras (cameras near the eyeball) mostly use global shutter CMOS imagers, due to their high frame rates, small motion blur artefacts and zero blooming effects, which are essential for eye tracking. The eye cameras performance heavily relies on the environmental conditions, introducing external noise. The image quality and accuracy of the pupil estimation heavily depends on the illumination conditions and optical axis alignment of the camera with the eyeball. An underexposed image will experience more visible noise since the signal is in close proximity to the base noise floor. An over-exposed image will simply get saturated and therefore carry no useful visual information. It is mandatory to account for the illumination condition by choosing an appropriate exposure time [60].

Besides illumination, the imager needs to be aligned with eyeball to get a good full view of the eyeball. This alignment must be maintained, if the pupil position is to be associated with a 2D or 3D image point from the scene camera, i.e. the pupil position is calibrated onto the world image. Every motion or deviation renders the calibration inoperable. This implies, that camera motion or slippage of the eye tracker is a critical external noise or uncertainty factor that needs to be accounted for. Slippage can be

countered either using an adaptive calibration algorithm or by other measures to prevent the headset from massive slippage, e.g. via an elastic headband, which is common practice in mobile eye tracking.

The scene camera image sensor is subject to the same internal and most external noise sources, depending heavily on the sensor's internal structure and technology. In contrast to the eye tracking camera, the scene camera will experience rather big changes of the visual scenery due to pose changes of the wearer's head between consecutive images. A global shutter camera is a must if the camera will experience fast motion changes. Otherwise, the image is subject to motion blur, effectively degrading gaze estimation accuracy during head motions.

## 2.3 Mathematical concepts of orientation representation

This section outlines the importance of quaternion math for navigation purposes and gives details about this. The chapter furthermore introduces navigation computation methods that are used in this work – they are divided into two subcategories – MARG only methods that do only rely on MARG sensor data fusion for orientation computation as well as camera vision supported MARG sensor methods that rely on additional camera hardware to calculate precise orientation estimation.

There are various mathematical methods to express orientation in three-dimensional space ( $\mathbb{R}^3$ ). This section explains the basic techniques for expressing an orientation of a body in three-dimensional space as well as the foundations for coordinate system transformations by means of rotations. The most prominent methods include the representation of an orientation using Euler angles or quaternions [41]. The latter are especially favored when it comes to the implementation of navigation or orientation estimation algorithms, due to their mathematical advantages over Euler angles within the implemented filter solution and because quaternions operate in  $\mathbb{R}^4$  which is easier to use for representation of orientation in  $\mathbb{R}^3$  [61]. Nevertheless, this section firstly explains Euler angle orientation representation, since it is easily understood as well as interpreted and will be used to present the estimation results and errors of the developed orientation estimation methods.

### 2.3.1 Preliminary conventions

This section includes the necessary information about coordinate systems and notation conventions that are crucial for understanding orientation representation in the  $SO^3$  group which are of substantial importance to understand the mathematical concepts used throughout this thesis.

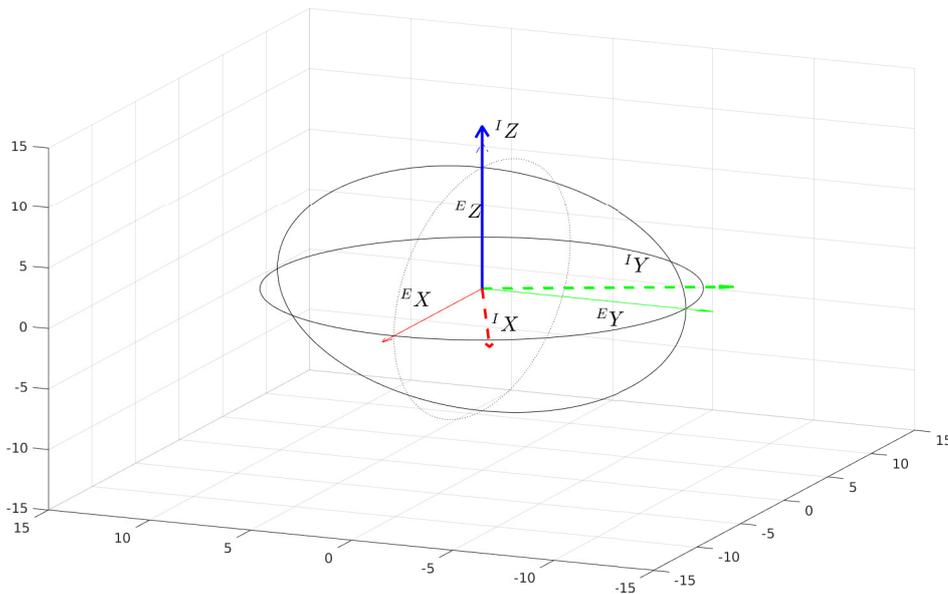
### Coordinate systems

Rotation denotes the relative orientation between two coordinate systems. For the navigation purposes, some standardized coordinate system conventions have been implemented, which will be explained in detail below.

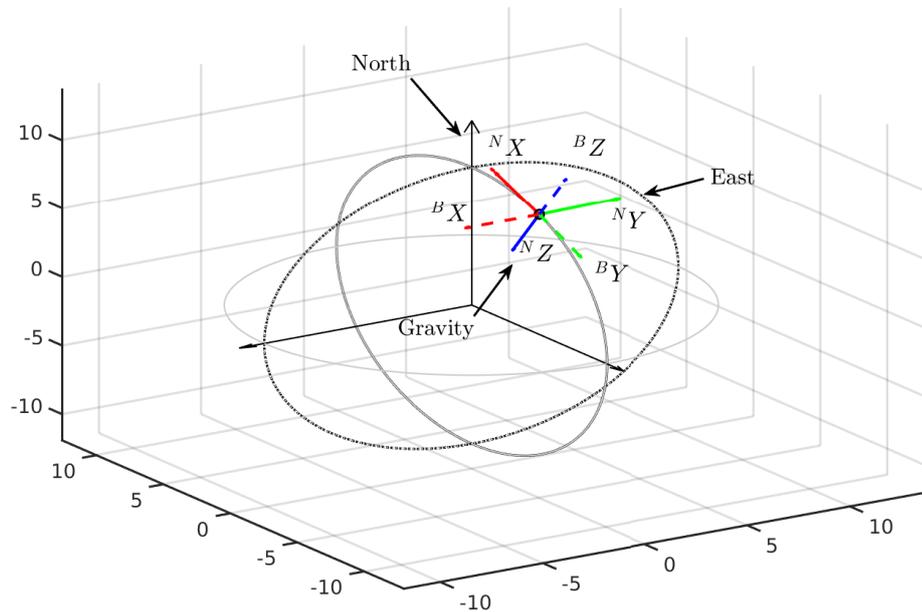
**Inertial frame [I]** The inertial frame is located at the origin of the rotational ellipsoid that resembles the shape of the earth, see figure 2.17. The x- and y-axis are fixed with respect to the fixed stars <sup>2</sup>. The z-axis coincides with the axis of rotation of the Earth's ellipsoid. Inertial sensors measure their quantities with respect to this coordinate frame (see accelerometer and gyroscope) [41].

**Earth frame [E]** The earth's coordinate system coincides with the origin of the rotational ellipsoid, see figure 2.17. The axes are fixed with respect to the globe and rotate

<sup>2</sup>fixed stars (stellae fixae) are stars that appear to be fixed in the night sky and always maintain the same position with respect to each other (compare star constellations of zodiac signs)



**Figure 2.17:** The rotational ellipsoid (earth) and the two coordinate systems, earth coordinate system (continuous lines, axis index E) and the inertial coordinate system (dashed lines, axis index I) are shown. Both coordinate systems coincide with the origin of the ellipsoid. The x- and y-axis of the earth coordinate system are fixed with respect to the ellipsoid and rotate about the z-axis with angular velocity  $\Omega$ . The inertial coordinate system is fixed with respect to the fixed stars.



**Figure 2.18:** The rotational ellipsoid (earth) and the two coordinate systems, navigation coordinate system (continuous lines, axis index **N**) and the body coordinate system (dashed lines, axis index **B**), are shown. Both coordinate systems coincide with the origin of the object. The x- and y-axis of the navigation coordinate system point to the north and east. The z-axis is parallel to the gravity. The body coordinate system is fixed with respect to the object.

around the vertical axis with the earth's angular velocity ( $\Omega$ ). This vertical axis represents the z-axis of the earth frame. This earth fixed coordinate system is especially necessary for near-earth navigation because an object usually moves fixed with respect to this coordinate system [41].

**Body frame [B]** The body frame describes the fixed coordinate system of a rigid body in three-dimensional space, see figure 2.18. This coordinate system consists of the fixed axes of the object whose position in space is to be determined by the MARG sensor. The sensor data is specified in the coordinates of this system if the axes of the MARG sensor coincide with those of the object. If the object and sensors coordinate system differ (e.g. in orientation), a transformation is required to align the MARG sensors sensitive axis (sensor coordinate system) with the objects coordinate system [41].

**Navigation-frame [N]** Within this coordinate system the navigation solution of the object itself is specified, see figure 2.18. The origin lies within the body coordinate

system. The x-axis points to the north pole of the earth while the y-axis points east. Both axes lie in the tangential plane of the rotational ellipsoid. The z-axis points parallel to the direction of the gravity vector [41].

#### Notation convention

Due to the existence of different coordinate systems for the description of an orientation in space or for the transformation of the measurement data from the different coordinate systems into the navigation coordinate system, a naming convention shall be used for the sake of clarity, which unifies the transformations of the individual quantities into the different coordinate systems. The following example will be used to introduce the chosen naming convention.

Consider the velocity vector  $\vec{v}$  of a sensor measured in coordinates of the body coordinate system, i.e. in the coordinates x, y, z of the body, which is indicated by the superscript  $B$ .

$${}^B\vec{v}. \quad (2.33)$$

However, this index does not give any information about which velocity is specified in this coordinate system. For this purpose, one other subscript index shall be appended to the vector. Within this example, it is the velocity of the body coordinate system with respect to the earth coordinate system as reference. That is, a body (sensor) with its body coordinate system moves with velocity  $\vec{v}$ , in units of the body coordinate system, with respect to the earth coordinate system [41].

$${}^E_B\vec{v}. \quad (2.34)$$

Thus, the subscript (here  $E$ ) gives information about the reference to the motion of a coordinate system. This notation makes it possible to represent the transformation of vectors or quaternions into another coordinate system understandably. For example, the velocity vector ( ${}^E_B\vec{v}$ ) from the above mentioned example is to be transformed into the navigation coordinate system. This is obtained using either quaternion multiplication of the vector with the corresponding quaternion or analogously by multiplying the vector with a direction cosine matrix (compare chapter 2.3.2 & 2.3.3) [41]. To denote the transformation the quaternion or the direction cosine matrix will be given two indices as well. In this example, the velocity vector  ${}^E_B\vec{v}$  is to be transformed into coordinates of the navigation coordinate system, which is indicated by the following indices:

$${}^N_B\mathbf{q}, {}^N_B C, \quad (2.35)$$

where  $\mathbf{q}$  denotes a quaternion and  $C$  a rotation matrix. Consequently, this indexing denotes the rotation  $\mathbf{q}$  of the body coordinate system ( $B$ ) with respect to the navigation coordinate system (here  $N$ ). This orientation is used to transform a vector from the first

coordinate system ( $B$ ) to the reference coordinate system ( $N$ ) and can thus be written as

$${}^N_E \vec{v} = {}^N_B \mathbf{q} \bullet \begin{pmatrix} 0 \\ {}^B_E \vec{v} \end{pmatrix} \bullet {}^N_B \dot{\mathbf{q}}, \quad (2.36)$$

where  ${}^N_B \dot{\mathbf{q}}$  is the conjugate quaternion, or by multiplying by the direction cosine matrix

$${}^N_E \vec{v} = {}^N_B C_E^B \vec{v}. \quad (2.37)$$

This notation will be used in the forthcoming work to describe transformations of the individual vectors and orientation representations into other coordinate systems. Global quantities, e.g. the gravity, are represented by a simple index (see equation (2.33)), because these quantities serve as reference for the description of the coordinate systems and thus cannot be indexed.

### 2.3.2 Euler angles

Euler angles are a set of three angles that can be used to describe the orientation of any object in three-dimensional space with respect to a fixed coordinate system. Euler angles were first introduced by Leonhard Euler in 1775. In aeronautics and engineering an alternative form has been developed, the so-called Tait–Bryan angles. This convention uses rotation around three distinct axes and will be used throughout this thesis. The angles ( $\phi$ ,  $\theta$ ,  $\psi$ ) each refer to a rotation of the object around a specific axis of the fixed coordinate system [62]. The angle  $\phi$  represents the rotation around the x-axis of this fixed system and is also referred to as roll angle within the navigation genre. The angle  $\theta$  represents a rotation around the y-axis and is referred to as the pitch angle. Lastly the angle  $\psi$ , which corresponds to a rotation around the z-axis is called the yaw angle (see 2.19) [41].

Euler angles are exceptionally good to visualize rotation and transformation in an understandable manner. The order of rotation is decisive for the orientation of the object and must therefore be strictly adhered to. Figure 2.20 illustrates an exemplary rotation sequence based on the initial coordinate system from Figure 2.19. The coordinate system is first rotated  $80^\circ$  around the z-axis ( $\psi$ ) and then  $70^\circ$  around the y-axis towards a new coordinate system.

The final rotation composed of all three single-axis rotations can be expressed as a single matrix and directly used to transform an object in terms of rotation. This matrix is the product of three independent rotation matrices for each axis respectively [62]. The rotation matrix can be directly described as a function of the Euler angles and is termed

### Euler Angle representation - Initial orientation

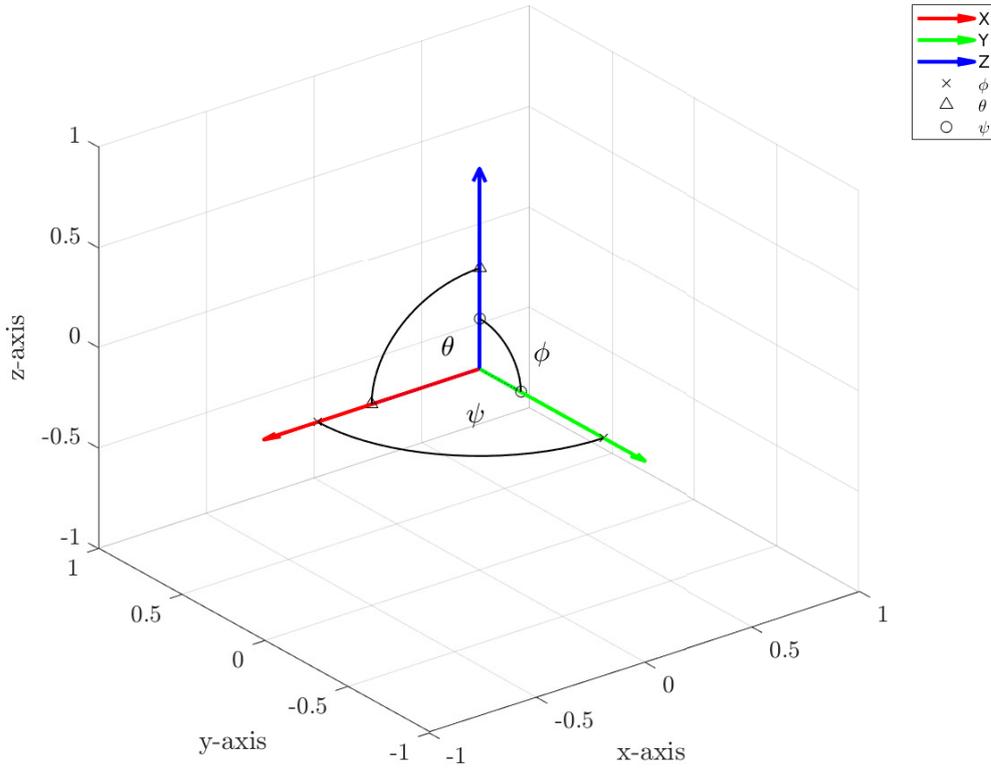


Figure 2.19: Depiction of the three Euler angles and their respective axis and planes of rotation.

Direction Cosine Matrix (DCM) and is given as follows

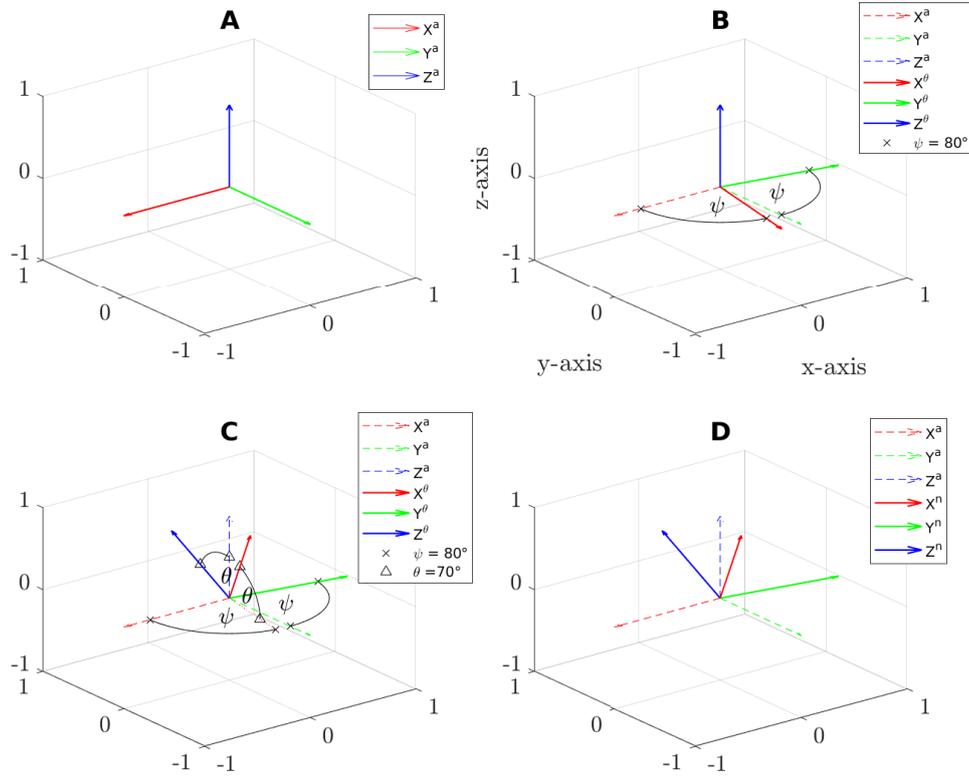
$${}^N_B C = \begin{pmatrix} c(\theta)c(\psi) & -c(\phi)s(\psi) + s(\phi)s(\theta)s(\psi) & s(\phi)s(\psi) + c(\phi)s(\theta)c(\psi) \\ c(\theta)s(\psi) & c(\phi)c(\psi) + s(\phi)s(\theta)s(\psi) & -s(\phi)c(\psi) + c(\phi)s(\theta)s(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{pmatrix}, \quad (2.38)$$

where  $c(x) = \cos(x)$  and  $s(x) = \sin(x)$  respectively. Successive rotations can be chained based on simple matrix multiplication.

This matrix can directly be applied to a vector transforming it into the desired orientation. This is done through simple matrix vector multiplication,

$${}^N x = {}^N_B C {}^B x. \quad (2.39)$$

Furthermore, we can find a set of Euler angles from a given DCM, which is exceptionally useful when converting between different orientation representations. The set of Euler



**Figure 2.20:** Example of a coordinate system transformation. Shown is an example coordinate transformation. **A** depicts the initial coordinate system (unit vectors)  $(X^a, Y^a, Z^a)$ . This coordinate system is first rotated by the angle  $\psi = 80^\circ$  (**B**) and results in the first rotation coordinate system  $(X^\psi, Y^\psi, Z^\psi)$ . This is followed by a rotation of the coordinate system from **B** by the angle  $\theta = 70^\circ$  (**C**) to the coordinate system **D**  $(X^\theta, Y^\theta, Z^\theta)$ . **D** depicts the final coordinate system after transformation  $(X^n, Y^n, Z^n)$  with respect to the initial coordinate system  $(X^a, Y^a, Z^a)$ .

angles from a given DCM can be calculated through the following

$$\phi = \arctan2(C_{32}, C_{33}), \quad (2.40)$$

$$\theta = \arcsin(-C_{31}), \quad (2.41)$$

$$\psi = \arctan2(C_{21}, C_{11}). \quad (2.42)$$

However, at certain configurations the rotation matrix to Euler angle mapping is singular and results in infinite solutions. This effect is called gimbal lock. Euler angles do suffer from gimbal lock if two axes of rotation align, resulting in singularities regarding the orientation expression. In the chosen rotation order (z,y,x) the gimbal lock will appear

### 2.3 Mathematical concepts of orientation representation

---

if the pitch angle (y-axis rotation,  $\theta$ ) will be greater than  $\pm 90^\circ$ . At this point, the cosine of  $\theta$  is 0 and the sine becomes 1. This leads to the following expression,

$${}^N_B C = \begin{pmatrix} 0 & -c(\phi)s(\psi) + s(\phi)s(\psi) & s(\phi)s(\psi) + c(\phi)c(\psi) \\ 0 & c(\phi)c(\psi) + s(\phi)s(\psi) & -s(\phi)c(\psi) + c(\phi)s(\psi) \\ -1 & 0 & 0 \end{pmatrix}. \quad (2.43)$$

In this configuration, the angles  $\phi$  and  $\psi$  always appear as a sum. Any independently chosen values for  $\phi$  and  $\psi$  will lead to a transformation with a fixed axis of rotation, while the only parameter that changes is the angle of rotation ( $\psi$  and  $\phi$ ). This only changes when  $\theta$  gets assigned another value [62].

To conclude, Euler angles are exceptionally good to visualize rotations in three-dimensional space but lack the ability to provide a continuous and singularity-free orientation representation. For this purpose, other representation methods exist.

#### 2.3.3 Unit-quaternion vector

A quaternion is a four-dimensional operator describing a rotation around a vector centered at the origin of the reference coordinate system [62]. A unit-quaternion vector is a normalized quaternion of length one. It consists of four elements and is denoted as  ${}^N_B \mathbf{q}$

$${}^N_B \mathbf{q} = (q_0 \quad q_1 \quad q_2 \quad q_3)^T = \begin{pmatrix} q_0 \\ \vec{q} \end{pmatrix}, \quad (2.44)$$

where  $\vec{q} = (q_1, q_2, q_3)^T \in \mathbb{R}^3$  and  $q_0 \in \mathbb{R}$ . A unit-quaternion is constructed through the adjunction of three hyper imaginary numbers ( $i, j, k$ ) to  $\mathbb{R}$ ,

$$\mathbf{q} = q_0 + iq_1 + jq_2 + kq_3. \quad (2.45)$$

A unit-quaternion vector lies within the unit-three-sphere  $\mathbb{S}^3$ . Each of these hyper-imaginary complexes can be assigned to a spatial dimension of the unit-three-sphere which can be seen from the mathematical connection to the axis-angle representation [62]. For a given rotation  $\theta$  around a unit vector  $\vec{e}$ , the unit-quaternion  $\mathbf{q}$  can be calculated as follows:

$${}^N_B \mathbf{q} = \begin{pmatrix} \cos(\theta) \\ \frac{e_x}{\theta} \sin(\frac{\theta}{2}) \\ \frac{e_y}{\theta} \sin(\frac{\theta}{2}) \\ \frac{e_z}{\theta} \sin(\frac{\theta}{2}) \end{pmatrix}. \quad (2.46)$$

The inverse of the rotation of a unit-quaternion vector is defined as the complex conjugate and is given as follows

$${}^N_B \dot{\mathbf{q}} = (q_0 \quad -q_1 \quad -q_2 \quad -q_3)^T = \begin{pmatrix} q_0 \\ -\vec{q} \end{pmatrix}, \quad (2.47)$$

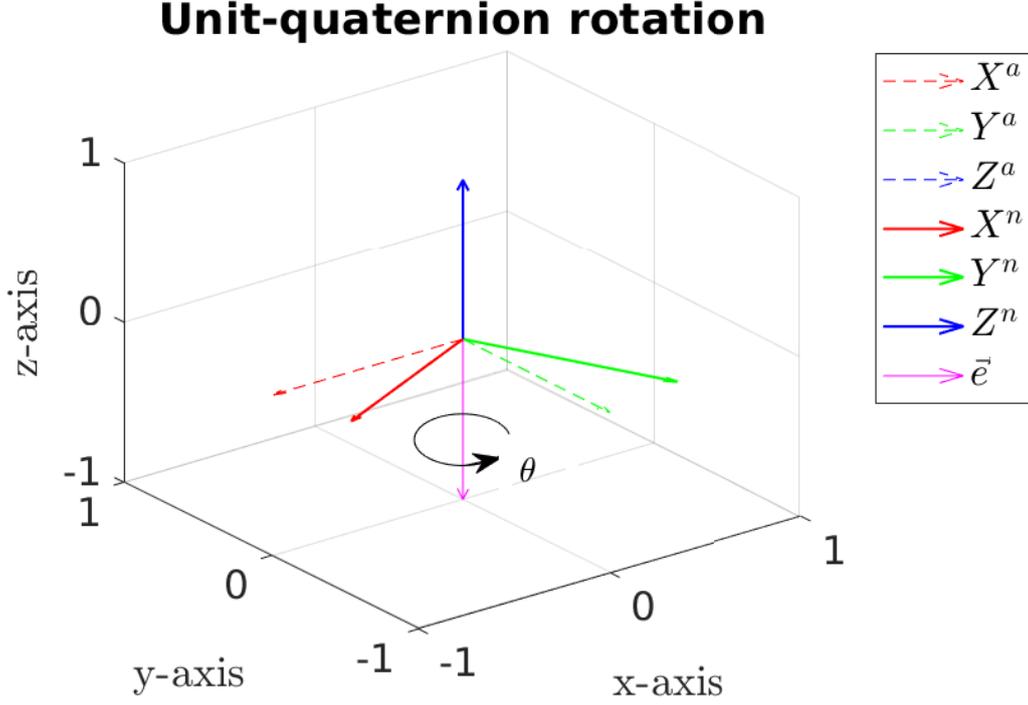


Figure 2.21: Unit-quaternion representation in 3D space.

where  $\dot{\mathbf{q}}$  denotes the conjugate unit-quaternion vector. Analogously to a rotation matrix, it is possible to link successive rotations using quaternion multiplication. The quaternion multiplication for a given set of quaternions ( $\mathbf{q}_1, \mathbf{q}_2$ ) can be expressed as a matrix vector multiplication, resulting in the following equation

$${}^N_B \mathbf{q}_1 \bullet {}^N_B \mathbf{q}_2 = \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix} \begin{pmatrix} q_{2,0} \\ q_{2,1} \\ q_{2,2} \\ q_{2,3} \end{pmatrix}, \quad (2.48)$$

where  $\bullet$  denotes the quaternion multiplication symbol. It should be noted that the quaternion multiplication is non-commutative [62]. This means that the order of multiplication is most important and must not be changed, e.g. when linking successive rotations. The neutral element of a unit-quaternion is the quaternion product of a quaternion and its conjugate and is defined as  $\mathbf{q}_I$ ,

$${}^N_B \mathbf{q}_I = {}^N_B \mathbf{q} \bullet {}^N_B \dot{\mathbf{q}} = (1 \ 0 \ 0 \ 0)^T. \quad (2.49)$$

Likewise to rotation matrices, quaternions can be used to directly rotate a given vector. The transformation is based on double-sided quaternion multiplication. The vector that is to be rotated is prepended with a zero and multiplied with the unit-quaternion vec-

### 2.3 Mathematical concepts of orientation representation

---

tor through equation (2.48). This new quaternion is now multiplied by the conjugate quaternion,

$${}^N \vec{v}_2 = {}^N_B \mathbf{q} \bullet \begin{pmatrix} 0 \\ {}_B \vec{v}_1 \end{pmatrix} \bullet {}^N_B \dot{\mathbf{q}}. \quad (2.50)$$

As shown in [62], it is possible to directly convert a unit-quaternion to a rotation matrix. After this conversion it is possible to extract a set of Euler angles. This conversion can be written as follows,

$${}^N_B C = \begin{pmatrix} (q_0^2 + q_1^2 - q_2^2 - q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & (q_0^2 - q_1^2 + q_2^2 - q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & (q_0^2 - q_1^2 - q_2^2 + q_3^2) \end{pmatrix}. \quad (2.51)$$

It is furthermore possible to calculate a unit-quaternion from a given set of Euler angles using the following formula

$${}^N_B \mathbf{q} = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} c(\phi/2)c(\theta/2)c(\psi/2) + s(\phi/2)s(\theta/2)s(\psi/2) \\ s(\phi/2)c(\theta/2)c(\psi/2) - c(\phi/2)s(\theta/2)s(\psi/2) \\ c(\phi/2)s(\theta/2)c(\psi/2) + s(\phi/2)c(\theta/2)s(\psi/2) \\ c(\phi/2)c(\theta/2)s(\psi/2) - s(\phi/2)s(\theta/2)c(\psi/2) \end{pmatrix}. \quad (2.52)$$

Unit-quaternions make it possible to describe and calculate a continuous and singularity-free orientation. Quaternions do not experience a gimbal lock and are therefore used to describe the orientation of an object in real-time within the navigation genre [62] as well as within this thesis.

#### 2.3.4 MARG sensor only orientation estimation methods

The precision, robustness and accuracy from MARG sensor orientation estimation depends on two main factors: a) The sensors hardware specifications (e.g. resolution, noise, and sensitivity) and b) the signal processing pipeline, also known as data fusion process, calculating the orientation estimate employing mathematical methods. The absolute ratings for the measurement capabilities of the sensor define the physical boundaries and represent as such hard limitations, while the data fusion method is a soft limitation. Different data fusion processes can lead to varying levels of accuracy when employed on the same set of sensor data [25].

Head-motion-based interfaces usually employ low-cost MARG sensors to estimate orientation without a need for static infrastructure, e.g. markers that would limit the useable motion range and environment [21]. The orientation estimation from these sensors is based on the angular rate integration measured by the gyroscope. This raw signal suffers from various noise terms, especially slowly-varying gyroscope bias and ARW that result in drift of the orientation estimation and therefore reduce accuracy. The drift is usually compensated using global reference vector measurements from accelerometer

and magnetometer [1], [22]. These correctional steps however experience limitations. The measurement of the gravity vector for example is achieved using a three-axis accelerometer. If the object is in motion, it will be subject to an external acceleration that will be measured alongside the gravitational acceleration and will ultimately result in a deviation of the gravity measurement vector. Moreover, the usage of the magnetic field measurement for heading corrections must be treated with caution due to ferro- or permanent-magnetic disturbances, especially in indoor environments [25].

Over the last years, much research effort has been put into the development of various filter methods to enhance accuracy and robustness and dealing with the afore-mentioned disturbances. Caruso et al. divide the majority of the filter approaches into two categories: Kalman filters and complementary filters [25].

**Kalman filter** Generally speaking, the Kalman filter is an adaptive filter which iteratively calculates a state estimate of unknown variables based on a series of noisy observations (or measurements). The Kalman filter consists of two steps. The state estimation, also known as prediction, and the update. In the prediction step, the filter uses the information provided by the previous time interval to make an a priori prediction based on a dedicated system model. In the update step, this estimate is corrected by a new weighted measurement or observation [41]. The weight given to the information provided by each measurement is governed by dedicated parameters based on the system and measurement noises, essentially the covariance. The higher the weight, the less the information provided is trusted, hence a Kalman filter is also known as inverse weighted filter [25]. The idea behind the Kalman filter is to calculate a more precise prediction of the current state by combining the predicted system state with a series of weighted measurements rather than using the noisy measurements only. For navigation purposes, a suitable state model must be developed representing or approximating an orientation estimation. This model should reflect the orientation of the object or MARG sensor, e.g. a quaternion or Euler angles respectively [63]. Based on the relationship between the measured values and the system state, different forms of the Kalman filters have been developed. For linear relationships, one uses the linear Kalman filter, non-linear correlations are solved using the extended Kalman filter, which essentially applies a non-linear system model and a linearization process of the covariance matrix at the propagation step. Regardless of the specific system model, the Kalman filter can be described by some general filter equations. The system has an observable state  $\vec{y}_k$  (the measurement) and an unobservable internal state  $\vec{x}_k$ . The state is estimated using a prediction algorithm (a priori) and updated in the next step using the internal measurement (a posteriori). The a posteriori estimation becomes the a priori estimation in the next iteration [41]. In order to estimate a state, the Kalman filter needs a system model of the observed system of the following form

$$\vec{x}_{k+1}^- = F_k \vec{x}_k + (B_k \vec{u}_k) + \vec{w}_k, \quad (2.53)$$

### 2.3 Mathematical concepts of orientation representation

---

where the current system state  $\vec{x}_k$  is passed to the state-transition model  $F_k$  describing the transformation from the current to the next state and  $B_k\vec{u}_k$  is a sometimes-needed term, describing influence of a control-input model  $B_k$  applied to the control vector  $\vec{u}_k$  onto the system. The last part of the equation introduces the system noise  $\vec{w}_k$ , which is used to account for uncertainties of the system model. It is assumed to be a zero mean, normally distributed white noise with covariance  $Q_k$ ,

$$\vec{w}_k \sim \mathcal{N}(0, Q_k). \quad (2.54)$$

The relationship between the system state  $\vec{x}_k$  and the measurement  $\vec{y}_k$  are given from the output equation

$$\vec{y}_k = H_k\vec{x}_k + \vec{v}_k, \quad (2.55)$$

where  $H_k$  denotes the measurement model and  $\vec{v}_k$  is the measurement noise, assumed to be a zero mean, normally distributed white noise with covariance  $R_k$ ,

$$\vec{v}_k \sim \mathcal{N}(0, R_k). \quad (2.56)$$

As it has been mentioned before, the Kalman filter can be separated into two steps: the prediction and the update:

**Prediction step:** The propagation of the a posteriori state vector ( $\hat{x}_k$ ) to the a priori state vector ( $\hat{x}_{k+1}$ ) is calculated based on equation (2.53), given as follows,

$$\hat{x}_{k+1} = F_k\hat{x}_k + (B_k\vec{u}_k). \quad (2.57)$$

Since the uncertainty of the state estimate rises with propagation, the estimate of the covariance matrix  $P_k$  needs to be adapted as well. The covariance matrix is a measure of the state estimation accuracy and is used to weight the measurement in the update step which improves the overall state estimate. In general, the covariance matrix for the state estimate is expressed as the variances of the state vector, where  $x_k$  is the true system state and  $\hat{x}_k$  the estimated state, resulting in

$$P_k = E \left[ (x_k - \hat{x}_k)(x_k - \hat{x}_k)^T \right], \quad (2.58)$$

where E is the expectation operator [41]. Using equations (2.58), (2.53) and equation (2.57) the a priori covariance matrix ( $\hat{P}_{k+1}$ ) can be written as:

$$\hat{P}_{k+1} = F_k P_k F_k^T + Q_k, \quad (2.59)$$

where  $P_k$  is the a posteriori estimate covariance matrix.

**Update step:** The update step applies the currently available measurements  $y_k$  and applies these onto the state estimation vector. For this purpose, the measured values ( $y_k$ ) are subtracted from the expected value of the measured values ( $\hat{y}_k$ ) of the measurement model. In the event that the expected value and the actual measured values are far apart, a significantly larger proportion of the actual measured values contributes to the correction of the state [41]. From equation (2.55), we can write the expected measurement value  $\hat{y}_k$  as

$$\hat{y}_k = H_k \hat{x}_k. \quad (2.60)$$

The expected measurement is subtracted from the actual measurement to form the innovation  $z_k$  which is used to update the state estimation,

$$z_k = y_k - \hat{y}_k. \quad (2.61)$$

The innovation is used to update the a priori estimate to get the a posteriori estimate. The update is done using a weighting matrix that essentially describes whether the measured values or the system dynamics should be trusted more. The weighting matrix (also known as Kalman Gain) is calculated as follows,

$$K_k = P_k^- H_k^T (H_k P_k^- H_k + R_k)^{-1}, \quad (2.62)$$

see [41] for a complete derivation. Using the Kalman Gain and innovation, the a posteriori state estimation can be calculated as

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - \hat{z}_k^-). \quad (2.63)$$

Finally, we update the estimate covariance matrix

$$P_k = (I - K_k H_k) P_k^-, \quad (2.64)$$

where  $I$  is an identity matrix.

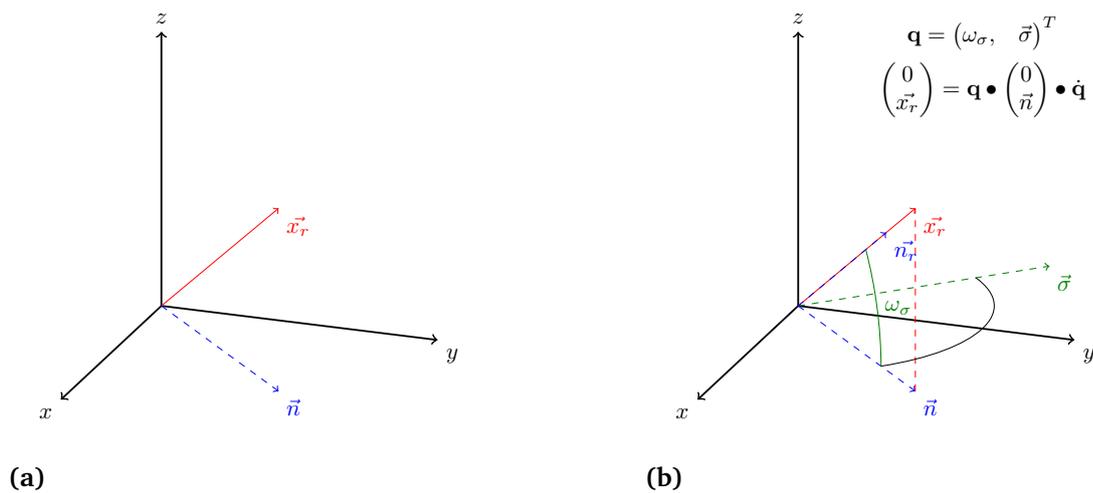
As has been mentioned before, the standard Kalman filter, as well as the extended or unscented Kalman filter follow the above-mentioned equations. The extended and unscented Kalman filter are used for non-linear state estimation problems. In the extended Kalman filter, the non-linear system model  $f$  and non-linear measurement model  $h$  are linearized using partial derivatives, transferring the system and measurement model into a transition- and measurement-matrix.

The second largest group of filters for MARG sensor orientation estimation are known as complementary filters. These filters are all based on the data fusion between two quaternions, where the first is generated from numerical integration of angular velocity measurements and the second quaternion is based on magnetometer and accelerometer measurements. The main difference between most complementary filters is the

### 2.3 Mathematical concepts of orientation representation

method used to calculate the correction quaternion from magnetometer and accelerometer measurements. The main advantage of complementary filters over Kalman filter approaches is the reduced complexity and amount of parameters and tuning of the filter and the considerable smaller computational load for self-contained systems, i.e. micro controllers. Complementary filters usually accommodate only one filter gain to tune the performance of the filter since they combine the measurement signals using complementary stages (high- and low-frequency filter combinations that complement each other). Especially one filter approach has gained popularity in both research and engineering fields due to the small size and ease of implementation on a microcontroller, namely the Madgwick filter.

**Madgwick filter:** The Madgwick filter is an optimization-based filter approach to compute orientation from MARG or IMU sensor data. It was developed by Sebastian O.H. Madgwick as part of his PhD thesis. It is designed to minimize both the necessary computation and implementation costs on a self-contained microcontroller system. It is based on the numerical integration of angular velocities expressed as a quaternion. The inherent gyroscope drift from the integration process is corrected using a weighted correction quaternion formed using the acceleration and magnetometer data. This correction quaternion is determined using a vector observation approach. The algorithm tries to find an orientation that aligns a predefined vector with a measured vector. This approach can be expressed as a minimization problem. A possible solution to this minimization problem is the gradient descent method (method of the steepest descent). A quaternion is calculated as a solution for acceleration vector as well as for the magnetometer vec-



**Figure 2.22:** Depiction of a quaternion multiplication that aligns an arbitrary vector  $\vec{n}$  with a measurement vector  $\vec{x}$ . Madgwick formulates it as a minimization problem to find the quaternion that satisfies this condition based on the Gradient-Descent Algorithm (GDA).

tor. Both quaternions are combined and merged with the quaternion obtained from the gyroscope data [36]. Chapter 4 gives more details about this filter.

Besides this filter, many more complementary filters have been proposed and extensively used. For example, Mahony's Filter [64], which is basically a PI controller calculating the error between the angular rate orientation and the accelerometer and magnetometer vector observations, or AQUA (algebraic quaternion algorithm) interpolating a quaternion, based on an algebraic derivation of a reference quaternion from accelerometer and magnetometer, with the quaternion from numerical integration of the angular rate using Spherical Linear Interpolation (SLERP) [65]. All of the above-mentioned filters do outperform the standard Kalman filter when it comes to computational performance, but can suffer from inaccuracies stemming from the different data fusion methods, e.g. not involving the different noise influences on sensor measurements and hence quaternion estimation [25].

Despite the different approaches to MARG sensor data fusion, Caruso et al. found, that the performance and accuracy for both types of filters heavily depend on the desired motion conditions and sensor hardware. In ideal conditions, both types of filters can perform more or less similar if they are tuned correctly for the desired motion or application case with average errors spanning from  $3.3 - 7.1^\circ$  [25]. The main advantages and disadvantages for both types of filter classes are summarized in table 2.2.

In conclusion, Kalman filters are superior when it comes to noise estimation and prediction without measurement updates, but are in need for high computational costs, due to the matrix inversion, and high tuning costs of the filter. Complementary filters, on the other hand, are easy to tune, have low computational costs (effective implementations) but the noise approximation and its influence on the estimate is mostly neglected. Based on the desired application, range of motion, surrounding external conditions (in-

**Table 2.2:** Comparison of Kalman vs. Complementary Filter.

<b>Kalman Filter</b>	<b>Complementary Filter</b>
High implementation costs	Small implementation costs
High computation time	Low computation time
More parameters (Tuning difficult but better modelling)	Small number of parameters (Less accurate model)
Optimal estimation *	Direct estimation
Variance based weighting	Gain based weighting <sup>+</sup>

\* For a linear system and measurement model and white system and measurement noise.

<sup>+</sup> Fixed or adaptive, usually not correlated to the probability function of the measurement

door/outdoor) and computational resources, an appropriate method has to be chosen. For further details about the state-of-the-art for complementary and Kalman filters for MARG sensors, the reader is referred to [25].

### 2.3.5 Visual orientation estimation methods

Video-based eye trackers are usually coupled with a scene camera to merge the pupil position prediction onto the present world scene view to get a grasp of what the user is looking at. Since this camera is in a fixed position with respect to the eye tracker frame and therefore the humans head, the camera enables the use of computer vision algorithms, such as optical flow, to estimate the ego motion of the camera. This estimation is widely known as Visual Odometry (VO).

Scaramuzza et al. describe VO as the three-dimensional reconstruction of the camera pose from subsequent images in real time [66]. VO methods follow the same general procedure and differ mainly in the individual methods used in the subsequent steps. Figure 2.24a depicts the general procedure for VO while figure 2.23 illustrates the general VO problem.

At first, an image is converted into grayscale (color is not important, only contrast is). The image is transferred to a feature detection algorithm to extract visual features, which are typically high contrast edges. The detected features are passed to a feature match-

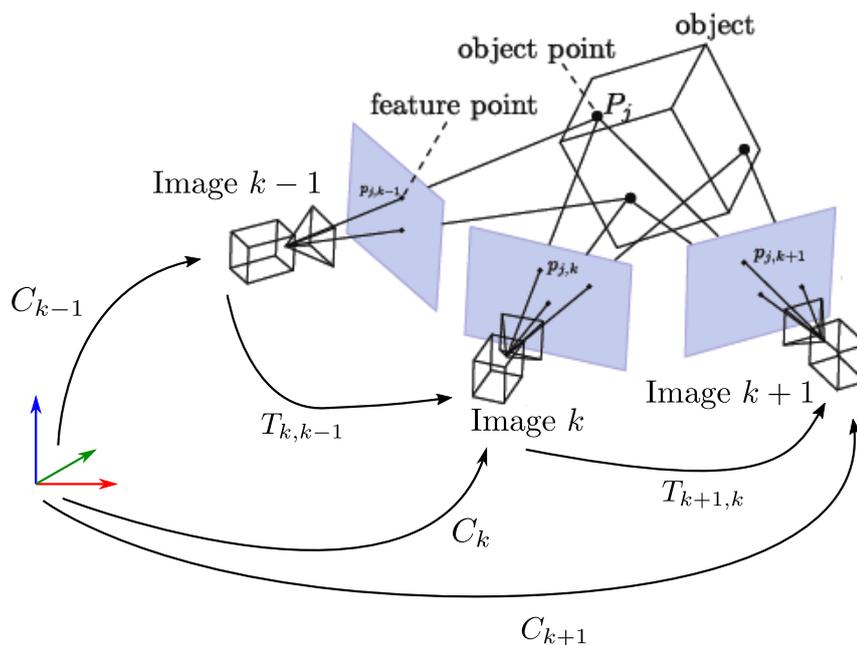


Figure 2.23: Illustration of incremental visual odometry estimation (adapted from [67]<sup>3</sup>).

<sup>3</sup><https://github.com/openMVG/openMVG/blob/develop/docs/sphinx/rst/openMVG/sfm/sfm.rst>

ing or tracking block that tries to match the features of the subsequent images (the previous frame) or tracks the features based on local search techniques, hence matching and tracking are two separate mechanisms. Two-dimensional features that represent the re-projection of the same 3D feature are called image correspondence and grouped together. These image correspondences are the foundation of the motion estimation. Based on the characteristic of the camera sensor used (monocular/stereo) a dedicated pose estimation has to be chosen (2D-2D or 3D-3D). Subsequent camera positions are related by the relative pose transformation  $T_{k,k-1}$  that is expressed as

$$T_{k,k-1} = \begin{pmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{pmatrix}, \quad (2.65)$$

where  $R_{k,k-1}$  denotes the rotation matrix and  $t_{k,k-1}$  the translation respectively. The relative pose estimations  $T_{k,k-1}$  of adjacent camera positions are computed from the detected visual features ( $p$ ). These poses are concatenated and usually passed to a local optimization approach, i.e. bundle adjustment, to calculate the absolute poses  $C_k$  w.r.t. to the initial coordinate [66]. Based on the type of features, monocular (2D) or stereo (3D), different motion estimation techniques need to be applied. In the monocular case, adjacent images are used to compute the essential matrix using the epipolar constraint. The essential matrix describes the geometric relationship between the two images. Consider figure 2.23. The adjacent images view a 3D object point from two positions. The epipolar constraint determines the line that enables the triangulation of the 3D object point  $P_j$  from the 2D image point  $p_{j,k-1}$  and image point  $p_{j,k}$  from two distinctive positions. The main disadvantage of monocular camera pose estimation is that depth is not observable from just a single camera, so we need to estimate a relative scale between adjacent images.

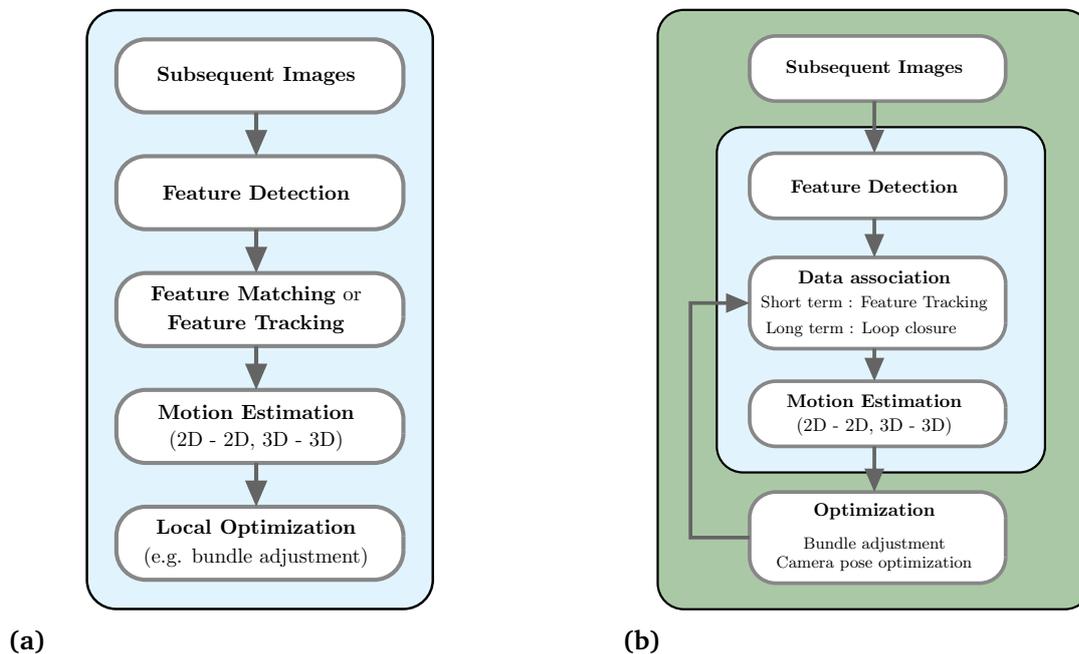
A stereo camera (3D) features tackles this problem right away. Both images can be used to triangulate 3D image points without the need to wait for adjacent frames.

Since VO is an incrementally computed camera pose, it also tends to drift due to accumulations of errors that arise between consecutive frames. Likewise to gyroscope integration, the pose integration will lead to an increasingly erroneous estimate. In pure VO cases, the drift can be reduced using a local optimization technique or using other global references, e.g. GPS or an IMU/MARG combination. Since VO is designed to incrementally compute a locally consistent pose, it is vulnerable to relative motion in the scenery. Consider the eye tracker scene camera that is used to track the pose of the human's head. A robotic arm might move through the scenery from left to right, while the user's head is stationary. If the feature points of the robotic arm cover a sufficient area of the image, any motion of the robot will be interpreted as motion of the head since it is not clear for the VO whether the robot or the head moves. Both result in a sufficient number of features moving across the visual scene. Besides this, VO is used for local consistency, meaning that if the visual data gets lost for a certain amount of time

### 2.3 Mathematical concepts of orientation representation

before the images are available again, the VO will integrate the pose from this point on forward, ignoring the blackout phase throughout and not being able to correct for this anyhow. In order to account for this and create a globally consistent pose estimation, visual SLAM approaches have been developed alongside pure VO.

A SLAM is used to generate a global consistent map and path that keeps track of the environment and the detected features in it to re-localize upon revisiting a known place [69]. SLAM approaches have been studied for the last decades, especially in the context of laser scanner-based SLAM. VSLAM has gained research interest recently due to the advances in CPU and GPU power to achieve real-time performance of the complex algorithms needed [69]. VSLAM does usually employ a VO algorithm to estimate the current pose and a loop closing module to create a consistent map when revisiting places, effectively reducing the accumulation drift, compare figure 2.24b. According to Scaramuzza et al. [66], the detection of loop closure and integrating these constraints effectively into the map are two main problems in the of vSLAM systems and of utmost research interest. vSLAM systems are more precise than VO but come at the cost of high computation load, complexity, and questionable real-time capabilities, but this depends on the use case and available hardware respectively. In the last decade, much effort has been put into the development of real-time and computational efficient vSLAM algorithms, due



**Figure 2.24:** General VO and V-SLAM illustrations. (a) illustrates the VO process steps (adapted from [66]) while (b) illustrates the process for pose graph optimization visual SLAM (adapted from [68]). The light blue boxes indicate VO building blocks while the green box shows the full SLAM process blocks. The loop closure module (inside data association) is not part of the VO process.

to the efforts taken for autonomous vehicles and robots leading to a boost in the usage of these algorithms in the scientific and engineering community.

There are basically two methodologies of vSLAM approaches, one uses probabilistic filtering techniques (e.g. Kalman filters or particle filters) to sequentially update probability distributions using all sensor measurements and the second one utilizes Bundle Adjustment (BA) methods performing batch optimization over key-frames (also known as pose graph optimization). Both methodologies are usually employed in a VO motion estimate over several images. The extended Kalman filter approach does marginalized out every pose except the current after every frame. BA on the other hand does discard almost all pose measurements, except for a view which are called keyframes. The latter BA group represents the most recent research approach. Both methodologies coexist and are further researched, but BA becomes more dominant due to the high efficiency regarding the computational costs for high accuracy [70]. For in-depth information about VO the reader is referred to [66] and [71], and for in depth details regarding vSLAM to [70], [72] and [73] respectively.

To account for drift and enhance orientation or pose estimation, recent approaches are combining MARG sensors with low-cost visual sensors, i.e. cameras. This type of data fusion process is known as Visual Inertial Odometry (VIO). Fusing these both sensing technologies not only enhances the accuracy for orientation estimations, due to the fact that more data and measurements are used, but furthermore enables accurate position estimation. Different data fusion methods have been researched and proposed over the last year. Most SLAM-based methods included an extended Kalman filter to fuse the data streams. Campos et al. recently published ORB-SLAM 3, a framework for visual- and inertial-based SLAM. They use an open-source description of visual feature (namely

**Table 2.3:** Comparison of VO vs. V-SLAM.

VO	V-SLAM
Small memory usage	High memory usage
Small computation load	Large computational load
Incremental pose estimation	Map + pose estimation
Drift from accumulation	Loop closure + relocalization counters drift*
Locally consistent	Globally consistent
Vulnerable to relative motion	More robust to relative motion

\* if map is sufficient and relocalization is successful

### *2.3 Mathematical concepts of orientation representation*

---

Oriented FAST And Rotated BRIEF (ORB)) to extract and track features [74]. Especially with the disruptive technological progress regarding computer vision and cameras, a wide field of data fusion methods has been developed to fuse the different data streams to enable a precise and robust pose estimation using MARG and visual sensors.



## 3 Infrastructureless head orientation estimation

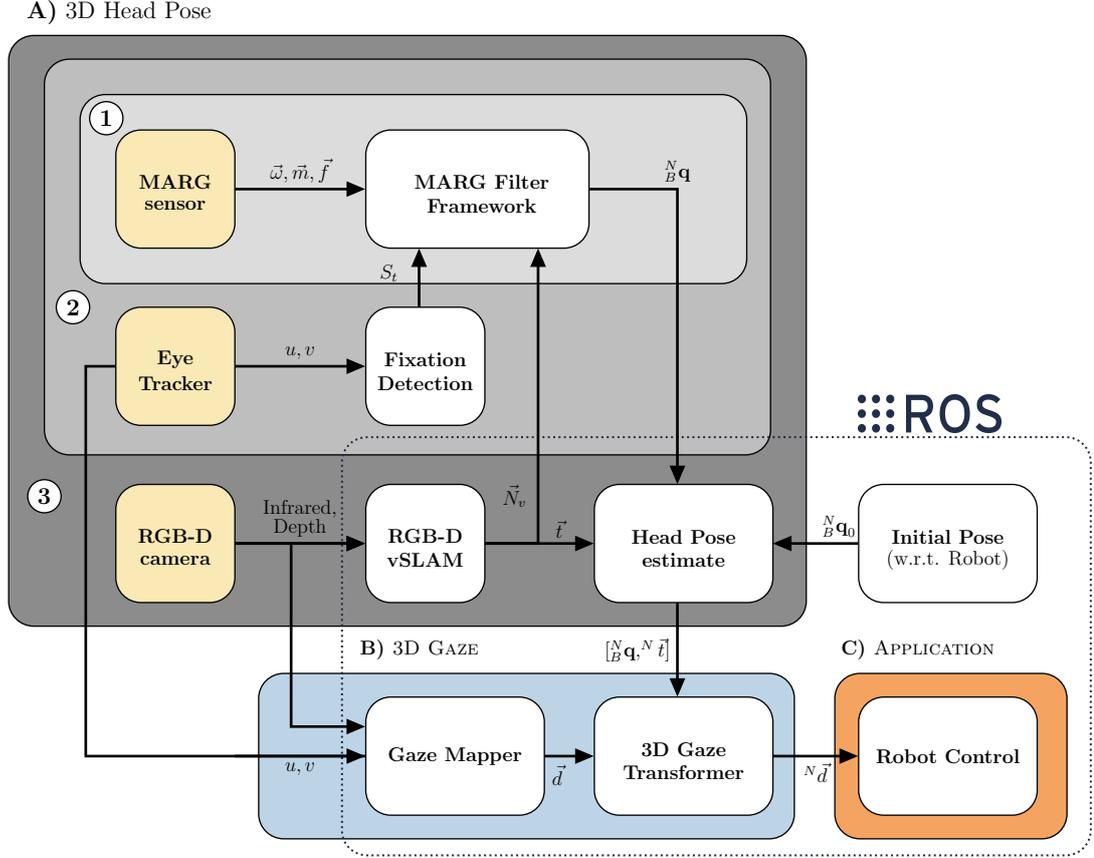
This chapter presents a simplified and general overview of the methods that, in connection with a head-worn interface, enable an infrastructureless and robust orientation estimation. The thesis presents three distinctive approaches with coherent methodology to achieve robust head orientation estimation. Each approach is an extension of the prior method fusing MARG sensor data with further sensor modalities to increase robustness during magnetic disturbances. The methods are complementary and can be used interchangeable depending on the desired accuracy or control mechanism. The underlying filter design incorporates quaternion math to bypass possible singularities if using Euler angle representations. The chapter furthermore presents the hardware and software specifications of the developed embedded MARG sensor system.

### 3.1 General concept

The basis for orientation estimation of this work lies within the filter framework of the MARG sensor. The framework was designed to deliver the basis of the orientation estimation pipeline to incorporate the three MARG sensor data streams and combine them into a reliable, singularity-free orientation expression. The linear quaternion design of the framework furthermore enables a fast and simple incorporation of other sensor signal measurements without the need for large adjustments or changes inside the filter equations. A switching mechanism is implemented that enables the filter to rate the heading references against each other and use the most reliable source. This feature is being used in a combined interface for three different heading sources: 1) the magnetometer heading reference, 2) a novel approach that incorporates eye tracking data to support the MARG sensor's heading estimate and 3) a vSLAM framework to compute a reliable heading reference to enable backwards corrections. Figure 3.1 depicts a simplified block diagram of the general concept.

To ensure this, the framework utilizes a two-stage filter approach.

**Block A) ①:** The thesis presents the idea to estimate orientation via a two-stage filter consisting of a gradient descent algorithm stage [36] and a linear Kalman filter to enable robust orientation estimation. The GDA filter stage computes a quaternion that is passed as the measurement input to a linear Kalman filter. This approach enables the following



**Figure 3.1:** Block diagram of the overall system setup. The system consists of three main software and hardware components that enable the calculation of robust control signals for robotic teleoperation: A) The accurate head pose estimation based on visual-inertial position and orientation estimation, B) the calculation of 3D eye and head gaze from known head poses and gaze points from dense 3D depth images as well as C) the application interface for robot control in 6D Cartesian space.

attributes: a) the computation of a continuous orientation as quaternion, b) short-time magnetic disturbance rejection, c) a weighted orientation fusion and d) exposing an interface for a variety of heading references without the need to adapt the filter equations and maintain accelerometer-based pitch and roll corrections.

The first filter stage features a gradient descent based-approach based on the work of Madgwick et al. [36] to calculate a measurement orientation from reference sensor measurements as a full quaternion solution. The second stage features a linear Kalman filter that incorporates the measurement quaternion from the GDA stage and fuses it with the gyroscope readings. This results in a full quaternion solution that calculates a complete orientation represented as quaternion and adaptively weights the estimation results based on the motion dynamics. Using a full quaternion as a measurement inside

the Kalman filter decouples the filter from specific sensor modalities and therefore from process model and equations. The upstream GDA filter on the other hand gives us the opportunity to include various reference modalities to form a full quaternion that can then be passed to the Kalman filter to allow robust pose estimation. This setup allows for robust orientation estimation while the magnetometer is calibrated, and the surrounding magnetic field stays mostly constant. To increase robustness, the filter is capable of switching towards an IMU type calculation mode if magnetic disturbances are present. In this mode, the filter rejects magnetometer readings and relies on gyroscope and accelerometer data only to overcome short-time magnetic disturbances. If the disturbances persist for a long period of time, the filter is not capable of delivering reliable orientation data and will be subject to errors over a longer period of time [1]. A detailed description is given in chapter 4.

**Block A) ②:** The second filter stage presents a novel approach that builds on top of the filter framework stage 1 and incorporates eye tracking as a new sensor reference modality for head orientation estimations while magnetic disturbance is present. The approach presented in this thesis utilizes the physiological correlation between pupil motion and head motion to detect stationary or dynamic motion states in order to support heading estimation of the filter framework. The detection of a stationary motion state is used inside the GDA filter stage to sample and hold the last orientation and extract a heading vector. This vector is used as the heading reference inside the GDA filter to reduce heading error accumulations while magnetic disturbance is present. The approach is implemented using a mobile eye tracker that is worn alongside a MARG sensor on the user's head. This approach does not need any eye to world calibration (also known as mapping), since it only depends on pupil position measurements in the eye-camera image and is therefore usable right out of the box. It enables the calculation of a full quaternion solution while magnetic disturbance is present and reduces heading error accumulations [4]. The error reduction rate depends on the number of stationary motion phases, but can be extended to compute an estimate of the current gyroscope bias to allow for less error accumulations over time. The approach is presented in chapter 5.

**Block A) ③:** The third approach presented in this thesis is the logical extension of the second approach. Since eye tracking glasses usually incorporate a scene camera to map eye-camera coordinates onto a world image stream, the scene camera can be used as a further reference sensor for orientation estimation [5]. The approach utilizes an open-source computer vision algorithm, namely ORB-SLAM 2 [75], to calculate the human's head pose. The orientation estimation is converted into a heading reference vector based on the process of the second approach. This vector is passed to the interface from filter stage 1 and used in the GDA filter to form a complete reference quaternion. This reference quaternion is fused inside the Kalman filter to generate reliable orientation

estimations. Combining the visual orientation estimation with a MARG sensor solves three main problems compared to the single sensing modality. First, the MARG sensor gives an initial orientation from global references which is not possible using only a vision sensor. Secondly, the MARG sensor enables the measurement of high dynamic motion and can correct motion blur artefacts from cameras during high dynamic motion, while a camera supports heading estimates during slow rotations. Finally, using a MARG sensor alongside a camera creates a countermeasure for relative motion drift from pure visual orientation estimation and thus creating robust orientation computations. Details about the procedure are given in chapter 5.

**Block B) and Block C):** Block B) enables a head or eye gaze based 3D Cartesian point estimation. Besides orientation support, the eye tracking glasses can be used as a further input modality for robot control. It gives a 2D gaze point estimate as well as fixation and blink detection for discrete or continuous control signals. Since the combined filter approaches (stage 1 to stage 3) compute a full head pose with respect to a common world frame, it is possible to enhance the eye gaze signal from a 2D image point to a 3D world point if the depth of the gaze vector is known. To allow for this, the approach utilizes the RGB-D cameras' depth image, which is used for gaze mapping. The head pose is used to transform a depth value from the depth image stream into the world coordinate system. The depth value can furthermore be chosen by either eye gaze or head gaze in the image stream. This 3D world point can be used to directly teleoperate the robotic EEF to this Cartesian waypoint. The robot application (Block C) finalizes the proposed concept for this thesis and incorporates the control methods for Cartesian point and 6-DoF robot control [5]. Details as well as experimental results can be found in chapter 6 and 7.

The combined approaches enable a robust orientation estimation under different disturbances: Approach A) ① enables short-time magnetic disturbance compensation by switching towards an IMU-based heading mode while Approach A) ② is able to reduce gyroscope accumulation errors during long time magnetic disturbances using visual fixations from eye tracking data. Approach A) ③ increases the orientation estimation robustness during long-term magnetic disturbances by using a heading reference from a visual SLAM approach. The approaches can be used in a complementary manner to compensate for visual artefacts and relative motion in addition to magnetic field interference. Based on the complete data fusion from A) ① - ③, Block B) enhances the input modality and delivers head or eye gaze Cartesian point robot control. The methods are described in detail in the following chapters.

## 3.2 Sensor system setup and design

The following section presents the modular developed hardware sensor system that functions as the baseline for a robust and infrastructureless head orientation estimation and further input modalities. The system consists of a self-designed MARG sensor board, a lightweight head-mounted eye tracker frame as well as an RGB-D camera module. The hardware blocks are interconnected based on modular 3D printed housing components that resemble the modular data fusion processes.

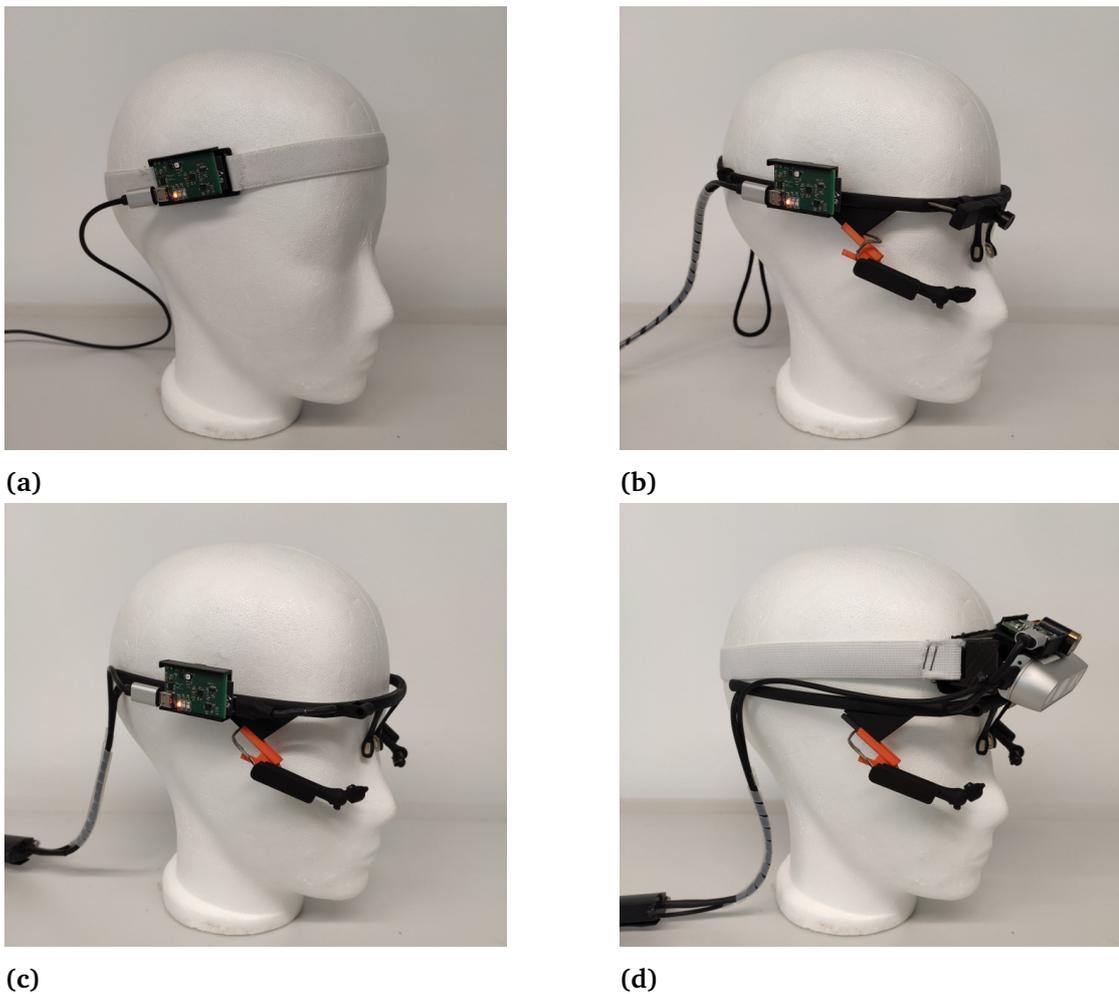
### 3.2.1 Modular system design

The sensor system consists of three main hardware parts that can be modularly interconnected based on the desired application layer, as can be seen on the general concept figure 3.1. It features a self-designed MARG sensor board (the WiOT system), an open-source lightweight head-mounted eye tracker (pupil core [54]) (monocular or binocular) equipped with an USB-C connector for an external camera, as well as an RGB-D camera (Intel RealSense D435 [76]), used as the scene camera of the eye tracker for gaze mapping. The system is fully modular and therefore extends the modular orientation estimation approaches presented in the general concept.

The WiOT system is a self-designed PCB equipped with a 32-bit dual-core microcontroller, a low-power 9-axis MARG sensor, a battery management circuit for Lithium-Ion Polymer (LiPo) batteries as well as communication interfaces for WiFi, BLE and UART communication. It will be referred to as WiOT in the ongoing chapter. The WiOT board is sitting in a 3D printed housing. The sensor is slid into the housing to circumvent the need for any (ferromagnetic) fixation screws. The housing has a fully open top side and 3 mm clearance between the bottom side of the MARG sensor and the housing to dissipate any heat that could build up inside a fully closed housing. The open top side furthermore allows the use of a heat sink on top of the microcontroller or MARG sensor to dissipate heat even further. The WiOT housing can be attached to an adapter plate via a dovetail bracket to be used within different modularization formats, see figure 3.2.

Based on the pose estimation method from either block (approach A) ①, A) ② or A) ③), the WiOT board can be attached to three distinctive adapter boards. In the pure MARG sensor only case, the board is attached to an adapter plate that can be worn like a headband, compare figure 3.2a.

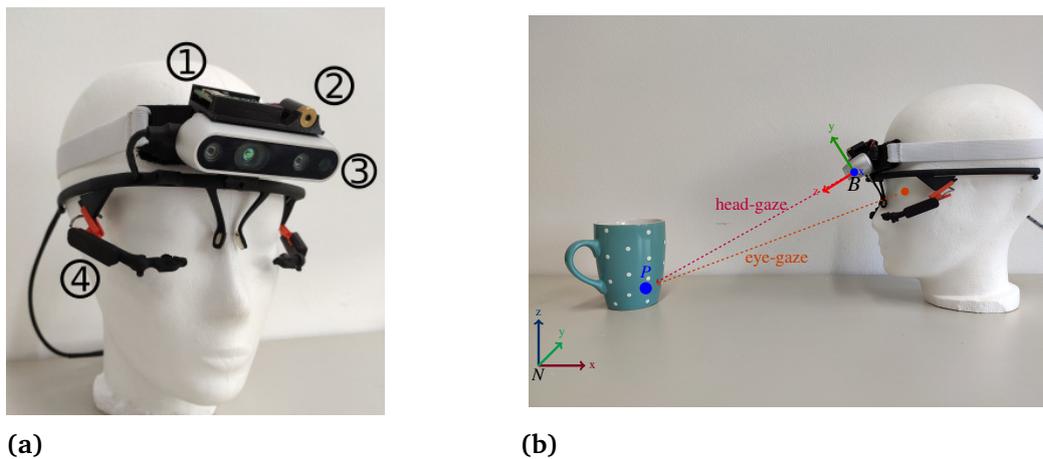
In the eye tracking-supported MARG sensor data fusion process, the sensor case is slid into an adapter plate that is fixated at one side of the eye tracker frame using a grub screw in such a way that both devices are rigidly interconnected. This is essential for the eye tracking and MARG sensor data fusion methodology that has been developed during this thesis and is presented in chapter 5. The tracker frame is secured via an



**Figure 3.2:** Modular sensor system design -(a) MARG sensor (WiOT), (b) MARG sensor attached to monocular eye tracking glasses (pupil core) using the dove tail adapter, (c) MARG sensor attached to binocular eye tracking glasses (without scene camera) using the dove tail adapter and (d) the MARG sensor attached to custom case carrying the RGB-D scene camera (Intel RealSense D435) on the binocular eye tracking frame.

eye-wear strap on the user's head. The adapter can be used with either the monocular or binocular eye tracker frame, see figure 3.2b-3.2c.

In the visual-inertial pose estimation case, the board housing is slid onto a 3D printed camera mount, carrying the RGB-D camera to ensure a rigid position transformation between the cameras and the WIOT's coordinate frame. The camera mount is angled downwards to ensure that the eye gaze mapping area covers a sufficient surface. It is equipped with tabs for a headband and features a notch that is placed over the glasses frame and fixated using a grub screw. The mount is secured via a headband on the user's head. This stabilizes the camera with respect to the eye tracker and distributes force from the nose rest of the tracker to the complete circumference of the head, therefore

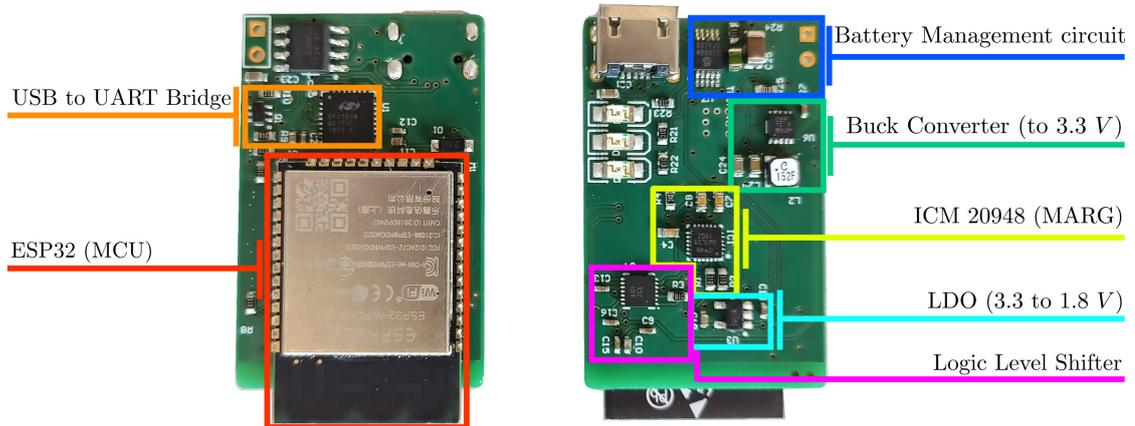


**Figure 3.3:** Proposed head interface (a) and depiction of head and eye gaze vectors (b). The head interface (a) consists of a pupil core binocular USB-C mount headset ④ and the custom camera mount uniting the depth camera ③, MARG sensor ① and feedback laser ②. Image (b) depicts head and eye gaze vector origins from the interface to a world point P.

reducing slippage from the eye tracker and increasing wearing comfort. Besides the sensor board, the camera mount holds a laser pointer which is aligned with the second infrared imager of the Intel RealSense camera. The laser pointer is used to give direct feedback for the user in robot collaboration tasks. The complete system weighs only 160 g. Figure 3.2d depicts a side view of the complete interface while figure 3.3 gives a more detailed overview of the sensor system and illustrates head and eye gaze rays in relation to the the system.

#### 3.2.2 The wireless orientation tracking system

The WiOT system is a self-designed low-cost consumer-grade MARG sensor board. It features a 32-bit microcontroller unit (MCU), a low-power 9-axis consumer-grade MARG sensor, the ICM 20948 [43], UART communication over Micro-USB, WiFi and BLE communication as well as a LiPo battery header to grant full wireless usage of the device. The MCU is an Espressif 32-bit dual-core microcontroller running the FreeRTOS real-time operating system. Figure 3.4 depicts a top and bottom view of the sensor board. The WiOT features a real-time and stand-alone implementation of the sensor data fusion process, running at up to 300 Hz and transmits orientation and raw sensor data at a constant rate of 100 Hz via Bluetooth Low Energy (BLE), WiFi or UART to a host target based on a binary User Defined Protocol (UDP). The system works in half-duplex mode and checks for received data in between every transmission. The receiving functionality includes parametrization, calibration as well as sending reference data that can be used inside the data fusion process (e.g. visual orientation estimation results). The following paragraphs give a detailed overview of the hardware and software components of the



**Figure 3.4:** The WiOT-system. Top and bottom view of the double-sided SMD assembly of the WiOT-system and the designated hardware blocks. Schematics of the system can be found in the Appendix.

WiOT system.

**Electromechanical specifications:** The WiOT system is a self-designed MARG sensor board that is capable of full wireless as well as wired operation mode. It consists of four main hardware parts:

- Espressif 32, a 32-bit dual-core MCU, 4 *mB* Flash ROM and on-chip antenna
- ICM 20948, a 9-axis MEMS MARG sensor featuring a low-level data processor
- A single-cell LiPo battery management circuit
- Serial communication Circuit Interface (UART) for flash and debugging

The board can be powered either directly through USB bus voltage (5 V) or through a 3.7 V header on the top left of the board. This header can be used to connect the system to a LiPo battery for full wireless operation. The board is also capable of charging a connected battery if it is connected to a 5 V USB source. In that case the battery management circuit charges a connected LiPo battery with a fixed 100 *mA* current at 4 V while powering the sensor board directly from the USB voltage, ensuring that the battery is not drained during the recharging cycle.

The sensor board utilizes three voltage rails due to the two different voltage range of the input sources, 3.7 V from LiPo and 5 V from USB, and operation voltage of the MARG sensor and MCU. The MCU is operated at 3.3 V which is provided using a step-down converter at the input stage of both the 5 V input source of the USB bus or the LiPo battery system. It operates from input voltages of 5.5 – 3.1 V, converting it to fixed 3.3 V DC with a maximum current supply of 200 *mA* (peak current). The ICM 20948 MARG

sensor operates at 1.8 V which is provided using a Low Drop-out (LDO) regulator circuit connected to the 3.3 V rail. The power electronics (5 V USB and LiPo input and battery management system) are connected to three low-voltage LEDs indicating the charging process, whether the battery is fully loaded and the supply voltage power status.

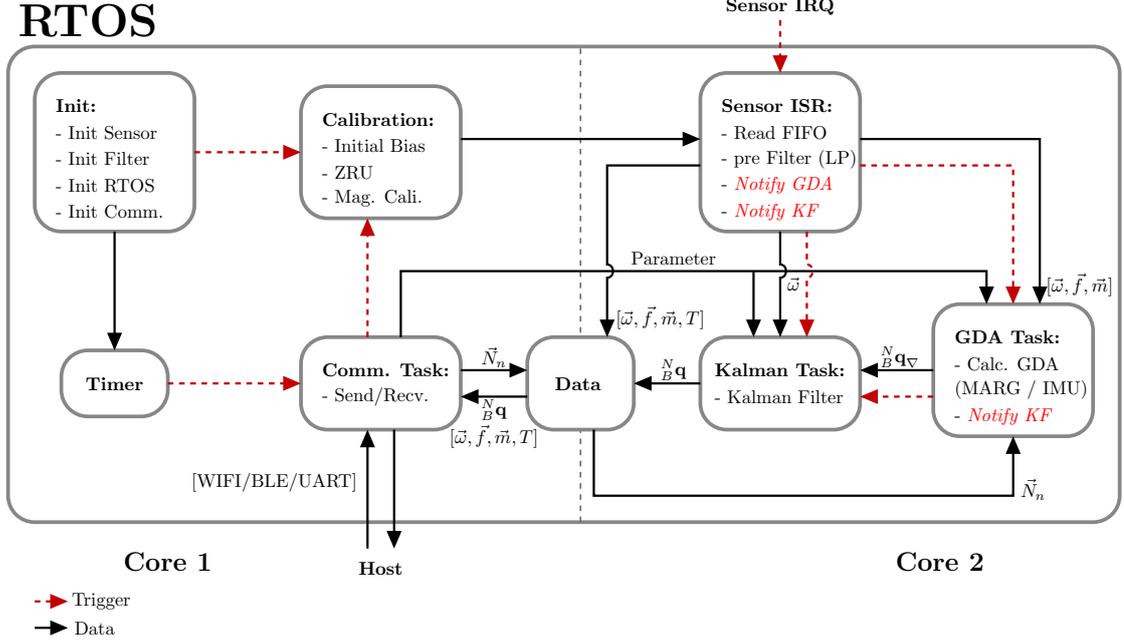
The ESP-32 is programmed using the UART TX and RX pins which are rooted to a serial communication IC converting UART to USB and vice versa. The board is equipped with a Surface-mount Technology (SMT) type B Micro-USB connector that has four Through-hole Technology (THT) pins for a mechanical robust connection. The THT mount provides a mechanical fixation point if a USB cable is plugged in.

**Software system & communication specifications:** The MCU of the WiOT system is running the FreeRTOS real-time operating system on both cores at 1 kHz scheduler tick rate. It features a stand-alone implementation of a quaternion-based Kalman filter that was developed during this thesis and is explained in detail in section 4. Using a real-time operating system gives outstanding advantages regarding timer-based callback registration, schedule the different callbacks with weighted priority and synchronize interrupts and tasks based on notification tokens.

The overall software concept can be divided into five dedicated subtasks and one initialization routine, which are either called once, called in a synchronous fashion based on a fixed function timer trigger or called asynchronous from a notification token due to an external sensor event. The tasks are as follows: initialization, sensor Interrupt Service Routine (ISR), calibration task, Kalman task, GDA task and a communication task. Figure 3.5 gives an overview of the task level interaction and implementation details. A complete and detailed description is provided below.

**Initialization:** The initialization routine has three main tasks. Initializing the FreeRTOS system with the dedicated tasks, notifications, and timers on both cores, checking and initializing the MARG sensor in order to trigger an initial bias calibration.

The launch sequence is triggered after an initial MARG sensor check. The ICM 20948 features a built-in functionality check that reports the sensor state using a single state flag. While no error is reported, the ICM 20948 is initialized with the desired sample rates and bandwidth for the gyroscope, accelerometer, and magnetometer. These parameters are parametrized and can be changed upon request. The sensor data ISR is registered onto Core two. After this, the calibration task is registered and notified to perform an initial start-up bias sample routine. The results are stored in the gyroscope and accelerometer registers respectively to prevent initial bias error accumulations. The communication task is registered onto core one and a 100 Hz timer is started, triggering the communication task. Lastly, the GDA and Kalman task are registered onto core one. After this, the initialization procedure switches into an idle state.



**Figure 3.5:** Block diagram of the developed software framework implemented on the ESP-32 microcontroller of the WiOT system. The functionality is embedded into an Real Time Operating System (RTOS), enabling multitasking and event-based task synchronization. The initialization, calibration and communication functionalities are implemented on the first core of the microcontroller, while the filter functionality is implemented on the second. This allows for a continuous communication with the host system regardless of latency of the filtered output. The red dashed arrows indicate a Trigger or notification signal, handling the task synchronization. The black arrows depict the data flow.

**Sensor ISR:** The sensor ISR is invoked upon a "FIFO has data" event, emitted from the Invensense ICM-20948 sensor. The ISR pulls the data from the ICM's internal FIFO and stores it in the ISR tasks stack. Based on the sample rates, the FIFO will most of the time only be filled with inertial sensor readings due to their superior sample rates of up to  $1\text{ kHz}$ . The magnetometer and accelerometer readings are sampled at  $100\text{ Hz}$ , while the gyroscope is sampled at  $300\text{ Hz}$ . If desired, the sensor data is pre-filtered, either using the ICM's digital on-board filter or a dedicated moving average filter. After the preprocessing stage, the data is pushed into the shared memory region, such that all tasks can access and use the data. The task furthermore inquires a temperature reading once per second and stores it in the shared memory region. Finally, the routine notifies the Kalman task upon any new gyroscope measurement and notifies the GDA task about every accelerometer or magnetometer update.

**Calibration:** The calibration task is used for initial and in-run bias estimation (the so-called Zero Rotation Update (ZRU)) of the inertial sensors and furthermore performs

the magnetometer calibration. The procedure is implemented on core one and triggered based on an asynchronous notification token. This token is given at start-up, if zero motion is detected for a continuous timespan (more than 5 s) or if it is explicitly invoked from the host (receiving a dedicated command Identifier (ID)). Since the calibration task is called asynchronously at very few times, it needs very few computational resources and is therefore implemented on core 1, residing next to the communication task. For the inertial sensor bias estimation, a dataset of 2000 samples are collected, low-pass filtered and averaged for each individual axis and stored at the sensor's bias register. Refer to section 2.1.2 and [77] for a high-level description of the calibration routine for the magnetometer.

**Communication task:** The communication task serves as a continuously triggered sending and receiving function. It is called using a 100 Hz timer token. Every token triggers a message generation and sending function. The data is fetched from a shared memory address and is cast into a single- or multi-message data package (64 or 128 bytes) using a binary UDP and passed to either a WiFi, BLE or UART hardware handling function. The communication mode needs to be defined before-hand and compiled onto the chip (WiFi or BLE functionalities desire a multitude of additional library files). After a successful data transfer (shifting bits into the output registers), the communication task blocks the task handler on core one to listen on incoming data packets. If a message arrives, it is decoded, interpreted, and executed. Incoming messages can either be direct system-level commands, e.g. to start a calibration, restart, change parameters, or receive sensor data from an external reference, e.g. from a vSLAM system or eye tracker. Due to the polling and blocking nature of this task it is implemented on core one, such that it does not interfere with the MARG sensor data acquisition or the filter calculations and furthermore enables a fast and half-duplex communication at a fixed rate. The fixed output rate is a crucial factor when dealing with real-time systems (e.g. robotic teleoperation), since the host target can determine the board's state from the fixed rate criteria.

**GDA task:** The GDA task calculates a complete measurement quaternion-based on an adapted version of Madgwick's gradient descent-based quaternion filter. It is notified by the sensor ISR to perform a full gradient calculation cycle if magnetometer and accelerometer data are available. Beforehand the notification ID is checked to determine which heading reference is available (e.g. magnetometer or external heading reference). If it features a reference, the filter stage calculates a complete reference quaternion from the accelerometer and heading reference. Note that the heading reference must be perpendicular to the plane defined by the gravity measurement (see chapter 4 for details). If the data does not feature a heading reference, the filter calculates an intermediate quaternion based on the last quaternion measurement and interpolates it based on gyroscope measurements. The filter calculates a complete but short-time substitute to the heading reference based on the process explained in chapter 4. This allows

for a complete calculation of a full measurement quaternion that will incorporate every accelerometer measurement at any time and fuse it with any heading source available, even if their sample rates diverge. The task notifies the Kalman filter that a full reference quaternion has been calculated. Besides this, the GDA Task is in charge of calculating a magnetometer confidence bound. This bound is calculated based on the dot product between different heading references and the Euclidean norm of the magnetometer readings. The confidence is used to determine whether the magnetometer data is trustworthy and should be included in the formulation of the measurement quaternion or discarded if it is disturbed. More details of this can be found in chapter 5.

**Kalman task:** The Kalman task calculates a full output quaternion using a linear set of equations combining the measurement quaternion from the GDA filter task and an ongoing prediction quaternion from angular-rate measurements. The Kalman task implements the well-known prediction and correction steps from a linear Kalman filter. Based on the availability of data, the filter is used in either closed or open-loop configuration. The task is notified either from the GDA task about a new measurement quaternion or from the sensor ISR to indicate the availability of a new gyroscope sample. If the notification stems from the ISR, the filter does a prediction only step with no update step and shifts the predicted quaternion into the shared memory region. This is the open loop configuration. If a measurement quaternion is available, marked by the GDA notification, the filter does a complete prediction and update step (closed loop). This updated quaternion is now moved to the shared memory region. This feature enables a mixed sampling rate data fusion. Furthermore, it allows for a high sampling frequency and prediction of orientation based on gyroscope data to reduce frequency-dependent noise terms and adequately capture the dynamic motion using gyroscope data, while correcting only if a full measurement quaternion is available.

The above-described software tasks are tightly coupled to the data fusion process that will be presented in chapter 4. Besides this filter, the WiOT software stack implements a stand-alone implementation of the Madgwick filter. Both filters can be used on the same data in parallel to the designed filter. By running these filters parallel on the same data, a direct comparison between the filter implementations is made possible.

The WiOT system is fully integrated into Robot Operating System (ROS), either wireless based on WiFi or wired using UART via a Micro-USB cable. The WiOT-ROS node is capable of handling multiple WiOT sensors through a broker node that listens on all incoming WiFi and USB messages on the host computer. Upon receiving an incoming "who am I" message on the broadcast address of the network, the broker broadcasts a request for the WiOT's unique chip ID. Upon receiving the answer of the WiOT, the chip ID is used to launch a new unique instance of a WiOT-ROS node to communicate with the sensor(s) and enable the bidirectional communication. To ensure wireless communication safety, the broker node requests the WiOT sensor to bind the communication

### 3.2 Sensor system setup and design

Table 3.1: Technical specifications of the WiOT system.

<b>Communication</b>	<b>Wired</b> UART (USB)	<b>Wireless</b> WiFi / BLE
<b>Sampling frequency</b>	<b>MARG sensor</b> Gyro. 300 Hz, Acc. 100 Hz, Mag. 100 Hz,	<b>Communication</b>  All interfaces - 100 Hz
<b>Sensor range</b>	<b>Typical</b> Gyro. $\pm 500^\circ s^{-1}$ Acc. $\pm 4 g$ Mag. fixed $\pm 4900 \mu T$	<b>Max. rating</b> $\pm 250$ up to $\pm 2000^\circ s^{-1}$ $\pm 2$ up to $\pm 16 g$
<b>Sensor ratings</b>	<b>Resolution</b> Gyro. $0.015^\circ s^{-1}$ Acc. 122 $\mu g$ Mag. 0.15 $\mu T$	<b>Noise density (datasheet)</b> $0.015 \frac{\circ s^{-1}}{\sqrt{Hz}}$ $230 \frac{\mu g}{\sqrt{Hz}}$ n.A.
<b>Latency</b>	<b>Data fusion</b> 300 Hz KF ( $\approx 1ms$ ) 100 Hz GDA	<b>Host sampling</b> Varying ( 1 ms)
<b>Power Consumption</b>	<b>Wired</b> max. $\approx 0.25 W^*$	<b>Wireless</b> max. $\approx 0.6 W^*$ (WiFi)
<b>Power Rating</b>	<b>Minimum</b> USB 4.7 V LiPo 3.7 V	<b>Maximum</b> max. 5.5 V max. 4.2V

\* @100 Hz. Communication frequency.

to a local IP address and port. This can be either the host computer itself or any other target computer in the network. The WiOT nodes open a name-specified topic that can be accessed within the ROS network. The broker node furthermore observes the incoming traffic in either the WiFi or the USB case to detect faulty communications. If a WiOT does not send data at a rate of at least  $100 Hz \pm 1 Hz$ , the unique WiOT-ROS node is suspended and restarted. This enforces a continuous and reliable communication for human-robot collaboration.

### 3.2.3 Mobile eye tracker - Pupil Core

The mobile eye tracker used in this work is a fully open source and extraordinary light weight head worn eye tracker. It is developed and distributed by a Berlin-based company, Pupil Labs [78]. Pupil Labs describes the company as a platform for the development of eye tracking systems and the related research areas. The basis of this platform forms the Pupil Core eye tracking headset that enables both monocular or binocular tracking of eye movement alongside a scene camera feed, either from a built-in high-speed camera or even a third-party camera.

The eye tracker captures the eye movement from near eye camera(s) at a resolution of  $192 \times 192$  px at a frequency of up to 200 Hz, or  $400 \times 400$  px at 120 Hz, using a dark pupil tracking method. Pupil Labs also provides an open-source software framework to control, calibrate, stream, and record eye tracking results. Pupil Labs reports an accuracy of  $0.6^\circ$  and a precision of 0.02 at ideal calibration conditions using the built-in calibration routines. The pupil framework consists of three different stand-alone software applications, of which two are important for the real-time interaction with the eye tracking glasses:

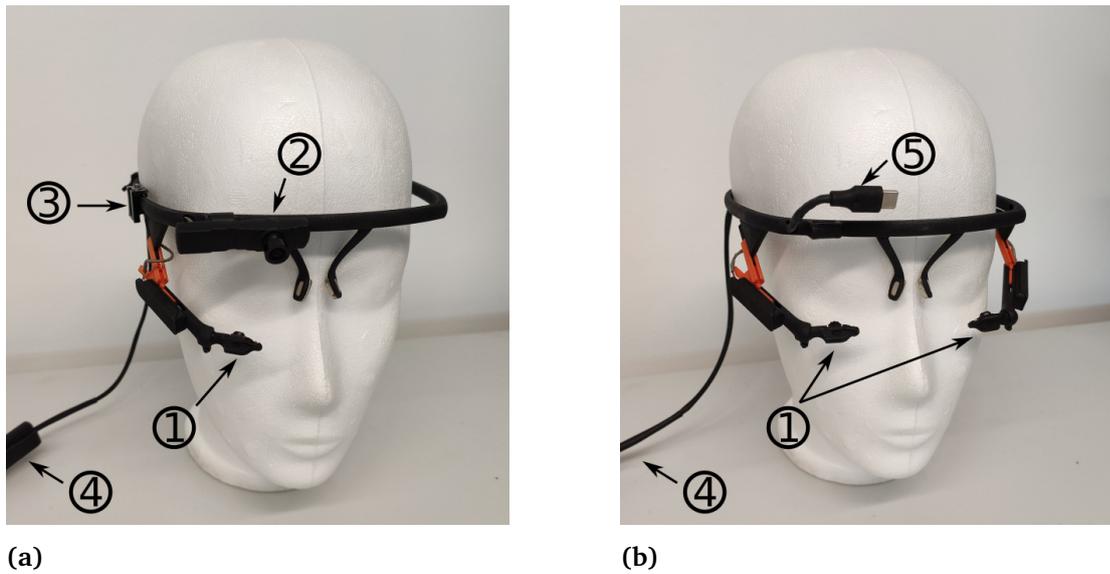
- **Pupil Capture** is the main software application for interfacing the Pupil Core headset. It reads the cameras' data streams, detects pupils, enables calibration of pupils to the scene camera streams (tracking gaze), records data and streams it over the Inter Process Communication (IPC) backbone.
- **Pupil Service** features the same functionalities as Capture but does not include a GUI. It is specifically designed to be integrated into Augmented Reality (AR) and Virtual Reality (VR) applications and is controlled using network commands to the IPC backbone only.

The eye tracker consists of a frame, a near-eye camera and a scene camera. The frame is 3D printed from Nylon PA12 material, which contributes to its extremely low weight. The tracker is connected to a host computer via USB 3.1 connection, the cameras (eye and scene camera) are wired through the tracker frame to a high-speed USB hub at the end of the glasses frame. The headset can be configured either monocular or binocular and with built-in scene camera or an USB Type C connector for a third-party camera, compare figure 3.6. This makes the Pupil Core extremely versatile for research and engineering purposes.

The performance of the eye tracking system is only partially defined by the hardware elements used. The open-source pupil capture software enables the operation of the eye tracker and evaluation of the results on all common operating systems (Ubuntu Linux, macOS, Windows 10), and furthermore on mobile applications (Android). The most important specifications of the Pupil Labs Pupil Core eye tracker are summarized in table 3.2.

A significant advantage of the Pupil Core eye tracking headset over current commer-

### 3.2 Sensor system setup and design



**Figure 3.6:** Pupil Core headset configurations: (a) Monocular eye camera ① with built-in camera ②, MARG sensor board adapter ③ and USB Type C female connector ④. (b) Binocular eye camera ① with USB Type C third party camera connector ⑤.

**Table 3.2:** Technical Specifications of pupil core eye tracker.

<b>Gaze accuracy</b>	<b>Accuracy</b> 0.6°	<b>Precision</b> 0.02
<b>Gaze parameter</b>	<b>2D Gaze</b> 2D gaze position	<b>3D Gaze</b> 3D gaze rays + 3D gaze point*
<b>Latency</b>	<b>Camera</b> 8.5 ms	<b>Processing latency</b> Varying with CPU ( 3 ms)
<b>Sampling frequency</b>	<b>Eye-camera</b> 200 Hz, 192 × 192px	<b>Scene camera</b> Built-in 120 Hz / 3rd. party
<b>Physical properties</b>	<b>Material</b> Nylon PA12	<b>Weight</b> 22.75 g <sup>+</sup>
<b>Interface</b>	<b>To Host Computer</b> Female USB Type C connector	<b>To scene camera</b> Male USB Type C connector

\* Binocular gaze ray vergence. Slippage compensation (to some extent) from 3D eye model.

+ without external camera.  
for binocular headset.

cially available models on the market is the fully open-source software and hardware approach. Not only is the tracking system designed in a way that it can be built in a "do it yourself" manner using off-the-shelf cameras, the open-source mindset is contin-

ued in their software framework. The software and algorithms are fully accessible and highly modular, facilitating community-driven customization and developments. The small size and low weight furthermore contribute to user-centered needs of the tracking glasses but naturally come at the cost of a lower overall accuracy compared to other available systems.

The fundamental advantage of the open-source Pupil software is the IPC backbone. The IPC exchanges relevant data between the individual processes of the system and defines an Application Programming Interface (API) that can be accessed by other software, e.g. ROS, on the same host computer or even by other hardware, e.g. another computer. The IPC backbone serializes and publishes the pupil, gaze, and image data. The published data can be accessed by different submodules or processes to perform calibration, pupil-to-gaze-point mapping and many more. The backbone exposes a ZeroMQ port that can be used by an external application to subscribe and read data directly from the IPC. This feature is used to directly access the data-streams in real-time using ROS, making all data streams (pupil data, gaze data, image data) available to a multitude of methods and plug-ins to directly integrate the eye tracker into robot control applications without blocking hardware ports.

## 4 MARG sensor based head orientation estimation

The content and results of the following chapter were published in [1] and are extended by further details.

The first and most important step to orientation estimation using MARG sensors is the underlying process of data fusion. There have been many research efforts over the last years regarding different MARG sensor data fusion strategies that focus on computational load, robustness to noise and small calculation errors. The Kalman filter approach with its various forms has become the most commonly used one and de facto standard for most applications. However, there have been other research attempts that were especially addressing computational load reductions on MCUs by using computationally reduced fixed-gain filters. These kinds of filters allow a small implementation size and fast calculations when compared with Kalman filters because these filters typically involve matrix multiplications which are a bottleneck when it comes to computational speed on a MCU. Kalman filters, however, have outstanding advantages when it comes to prediction of noise and weighting of measurement inputs.

During this thesis a data fusion process was designed that allows for increased robustness to overcome short-time magnetic disturbance compromising the heading estimate by estimating orientation via a two-stage filter consisting of a GDA stage and a Kalman filter framework (KFF), compare figure 4.1 for a simplified block diagram. It is done by switching from MARG- to IMU-type fusion and vice versa inside the gradient-descent filter stage (GDA). This stage employs a quaternion representation and is treated like a measurement inside the Kalman filter. The computation of only one quaternion from

A) 3D Head Pose

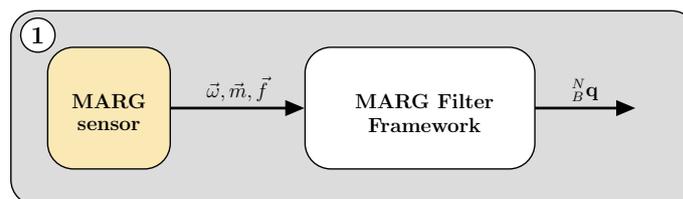


Figure 4.1: Simplified block diagram of the MARG filter fusion stage.

both the magnetometer and accelerometer data reduces the size of the needed model and state vectors. The Kalman filter incorporates the gyroscope model as quaternion and calculates a final orientation, represented as quaternion. The computation of the orientation inside the Kalman filter is weighted with respect to noise variance from motion and allows for a more accurate solution based on the sensor characteristics. Separating both filters in this fashion furthermore accounts for sampling frequency differences. A magnetometer can typically be read with a lower sampling frequency compared to a gyroscope or accelerometer. The gyroscope can be used to update the filter's state, even if no new measurement is available. This is also the case if any other heading (yaw) reference measurement variable is introduced to the GDA stage that might have a lower sampling frequency.

## 4.1 A quaternion-based filter framework

The two-stage filter approach is designed to run asynchronously to emphasize and benefit the different sample rates of the gyroscope, accelerometer, and magnetometer. The filter is designed to fuse a measurement quaternion from a preceding gradient-descent filter with a prediction quaternion inside a linear Kalman filter. Using a full quaternion as measurement input enables the usage of any possible measurement quaternion as the input signal and reduces the complexity of multiplications, effectively reducing the computational load. The preceding gradient-descent filter essentially enables the calculation of a reference quaternion based on the global quantities gravity and geomagnetic field while the computation load is kept at a minimum. The gradient stage is designed to switch between a MARG- and IMU-mode, ensuring orientation estimation while magnetic disturbances are present. The filter switches towards the IMU mode to discard magnetic measurements for the heading estimate and use gyroscope estimations instead to compensate short-time magnetic disturbances. The complete filter development is discussed in the following sections.

### 4.1.1 Orientation from angular rate

A gyroscope measures the angular rate that is applied to it in its body frame. The quantity (angular rate) is denoted as  $\vec{\omega}$ , emphasizing the vectorial notation to indicate the three distinctive measurement axes  $x$ ,  $y$  and  $z$  respectively. The angular rate describes the relative change of orientation with respect to one measurement. This can be represented as a quaternion ( ${}^N_B \mathbf{q}$ ) describing the rate of change regarding the orientation of the global navigation frame (index  $N$ ) relative to the body frame (subscript  $B$ ) [79] and can be written as

$${}^N_B \dot{\mathbf{q}}_{\omega, k+1} = -\frac{1}{2} \begin{pmatrix} 0 \\ {}^B \vec{\omega}_k \end{pmatrix} \bullet {}^N_B \mathbf{q}_k \quad (4.1)$$

## 4.1 A quaternion-based filter framework

---

where  $\bullet$  denotes the quaternion multiplication and  ${}^B\vec{\omega}_k$  is the angular velocity vector of the three-axis arranged

$${}^B\vec{\omega}_k = \begin{pmatrix} \omega_{x,k} & \omega_{y,k} & \omega_{z,k} \end{pmatrix}^T. \quad (4.2)$$

Under known initial conditions the angular velocity may be integrated over time to compute the sensor's orientation relative to the navigation frame. This orientation  ${}^N_B\mathbf{q}_{\omega,k}$  can be computed by numerically integrating the quaternion derivative  ${}^N_B\dot{\mathbf{q}}_{\omega,k}$  from (4.1) and given initial conditions

$${}^N_B\mathbf{q}_{\omega,k+1} = {}^N_B\mathbf{q}_k + {}^N_B\dot{\mathbf{q}}_{\omega,k}\Delta t \quad (4.3)$$

The quaternion would in theory be sufficient to track orientation. As has been stated in section 2.1.2 gyroscopes, especially MEMS gyroscopes, are subject to a variety of noise (ARW, DC bias, etc.) that lead to a drift in the orientation estimate from integration. To account for this, global reference measurements are used, that a) enable an initial orientation estimation and b) account for accumulation of sensor drift by correcting the orientation estimate. In the case of a single MARG sensor, the reference vectors are the geomagnetic field and the gravity vector. These vectors are used to compute a reference orientation represented as quaternion.

### 4.1.2 Orientation from the earth's magnetic and gravity field

As proposed by Madgwick et al. [36], one can estimate the reference orientation of a MARG sensor  ${}^N_B\mathbf{q}$  based on vector observation. The orientation of the sensor relative to the global navigation frame can be found as a quaternion which transforms a predefined reference field with known direction  ${}^N\vec{d}$  (i.e. gravity or magnetic field) into the measured direction of this field in the sensor frame  ${}^B\vec{s}$ . There will be an infinite number of possible orientation solutions if using a single vector measurement only. This might be sufficient for an insufficient solution using Euler angles (e.g. roll and pitch). However, a quaternion representation requires a single solution which leads to the formulation of an optimization problem [36]

$$\min_{{}^N_B\mathbf{q} \in \mathbb{R}^4} f({}^N_B\mathbf{q}, {}^N\vec{d}, {}^B\vec{s}) \quad (4.4)$$

where

$$f({}^N_B\mathbf{q}, {}^N\vec{d}, {}^B\vec{s}) = {}^N_B\mathbf{q} \bullet \begin{pmatrix} 0 \\ {}^N\vec{d} \end{pmatrix} \bullet {}^N_B\dot{\mathbf{q}} - \begin{pmatrix} 0 \\ {}^B\vec{s} \end{pmatrix}, \quad (4.5)$$

where  $\dot{\mathbf{q}}$  represents the quaternion conjugate.

Madgwick et al. proposed the GDA as a possible solution to this optimization problem because of its low implementation and computational costs [36] leading to the general

form

$${}^N_B \mathbf{q}_{k+1} = {}^N_B \mathbf{q}_k - \mu_k \frac{\nabla f({}^N_B \mathbf{q}_k, {}^N \vec{d}_k, {}^B \vec{s}_k)}{\|\nabla f({}^N_B \mathbf{q}_k, {}^N \vec{d}_k, {}^B \vec{s}_k)\|}, \quad k = 0, 1, 2, \dots, n \quad (4.6)$$

where  $\mu_k$  is the step size of the GDA and  $\nabla f$  denotes the gradient defined as the product of the objective function  $f$  and its Jacobian  $J$

$$\nabla f({}^N_B \mathbf{q}, {}^N \vec{d}, {}^B \vec{s}) = J^T({}^N_B \mathbf{q}, {}^N \vec{d}) f({}^N_B \mathbf{q}, {}^N \vec{d}, {}^B \vec{s}). \quad (4.7)$$

The equation simplifies if the direction of the reference fields is assumed to only have components in one or two major axes of the global frame. Following Madgwick's convention this leads to the assumption that the gravity field defines the vertical z-axis, and the magnetic field only has components in the x- and y-axis due to the inclination of the magnetic field. This breakdown results in two objective functions and their Jacobian for gravity and the magnetic field respectively [36]. Depending on the desired application, either attitude and heading for MARG sensors or just attitude for IMU sensors,  $\nabla f$  has to be chosen. Substituting  ${}^B \vec{s}$  for normalized accelerometer ( ${}^B \vec{a}_k$ ) or magnetometer measurement ( ${}^B \vec{m}_k$ ) and  ${}^N \vec{d}$  for either normalized gravity or normalized magnetic field vector ( ${}^N \vec{b}$ ) results in a set of equations to form the gradient of function  $f$  ( $\nabla f$ ) [36]

$$\nabla f_a = J_a^T({}^N_B \mathbf{q}_k) f_a({}^N_B \mathbf{q}_k, {}^B \vec{a}_k) \quad (4.8)$$

$$\nabla f_{a,m} = J_{a,m}^T({}^N_B \mathbf{q}_k, {}^N \vec{b}), f_{a,m}({}^N_B \mathbf{q}_k, {}^B \vec{a}_k, {}^N \vec{b}, {}^B \vec{m}_k) \quad (4.9)$$

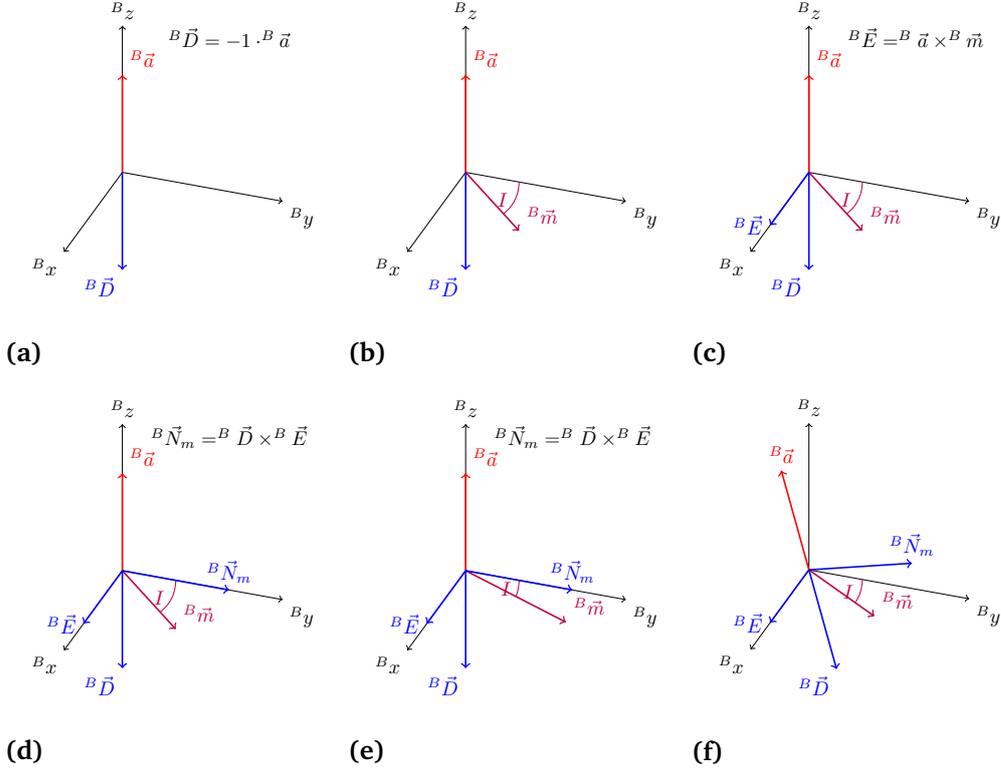
where the subscript  $a$  indicates equations for IMU and  $a, m$  for MARG sensor mode. Equation (4.9) denotes the simplified equation of the two gradient functions for magnetometer and accelerometer measurements. For the sole purpose of repeatability and readability, the full equations for the gravity- and magnetic field-based objective function from equation (4.5) and their respective Jacobians for MARG and IMU case are given below. The gravity- and magnetic field-based objective functions can be significantly reduced if the reference direction vector  ${}^N \vec{d}$  has only one non zero component [36]. For the acceleration function,  ${}^N \vec{d}$  can be assumed to have only a component in the z-axis. For the magnetic field vector, one can employ a horizontalization procedure to reduce the magnetic field vector to only a single component in the x-axis. This vector is called the north direction vector. It is computed using a set of cross products from accelerometer and magnetometer measurements to enforce a perpendicular vector and compensate the inclination of the magnetic field measurement. Figure 4.2 depicts the process of computation graphically.

The north direction vector is defined as the cross product between the down and east vector,

$${}^B \vec{N}_m = {}^B \vec{D} \times {}^B \vec{E}, \quad (4.10)$$

where the down vector is defined as the inverse of the acceleration measurement vector

## 4.1 A quaternion-based filter framework



**Figure 4.2:** Depiction of the north direction vector calculation from accelerometer (red) and magnetometer (magenta) readings. The resulting north direction vector is independent of the inclination from the magnetic field measurement (see figures e and f) and will consequently reduce the objective and gradient function.

$$({}^B\vec{a}) \quad {}^B\vec{D} = -B_{\vec{a}}, \quad (4.11)$$

and the east vector is defined as the cross product between the down vector and the magnetometer measurement vector ( ${}^B\vec{m}$ )

$${}^B\vec{E} = B_{\vec{D}} \times B_{\vec{m}}. \quad (4.12)$$

The north direction vector only has a single component since it defines the x-axis of the reference system and is not subject to the inclination angle.

The reference direction of the gravity and magnetometer direction vectors can be expressed as:

$${}^N\vec{d}_a = (0 \ 0 \ 1)^T, \quad {}^N\vec{d}_m = (1 \ 0 \ 0)^T. \quad (4.13)$$

Using equation (4.13) in (4.6) the objective function for the gravity vector can be written

as:

$$f_a({}^N_B \mathbf{q}_k, {}^B \vec{a}_k) = \begin{pmatrix} 2 \cdot (q_0 q_2 - q_1 q_3) - {}^B f_x \\ 2 \cdot (-q_0 q_2 - q_1 q_3) - {}^B f_y \\ 2 \cdot (-q_0^2 + q_1^2 + q_2^2 - q_3^2) - {}^B f_z \end{pmatrix}, \quad (4.14)$$

With (4.14), the Jacobian can be expressed as the following:

$$J_a^T({}^N_B \mathbf{q}_k) = \begin{pmatrix} 2q_2 & -2q_3 & 2q_0 & -2q_1 \\ -2q_1 & -2q_0 & -2q_3 & -2q_2 \\ -2q_0 & 2q_1 & 2q_2 & -2q_3 \end{pmatrix} \quad (4.15)$$

where  $q$  refers to the quaternion from the previous timestamp and the index denotes to the quaternion part.

The objective function and Jacobian for the magnetic field vector observation result in

$$f_m({}^N_B \mathbf{q}_k, {}^B \vec{m}_k) = \begin{pmatrix} (q_0^2 + q_1^2 - q_2^2 - q_3^2) - {}^B m_x \\ 2 \cdot (q_1 q_2 - q_0 q_3) - {}^B m_y \\ 2 \cdot (q_1 q_3 - q_0 q_2) - {}^B m_z \end{pmatrix}, \quad (4.16)$$

and

$$J_m^T({}^N_B \mathbf{q}_k) = \begin{pmatrix} 2q_0 & 2q_1 & -2q_2 & -2q_3 \\ -2q_3 & 2q_2 & 2q_1 & -2q_0 \\ 2q_2 & 2q_3 & 2q_0 & 2q_1 \end{pmatrix}. \quad (4.17)$$

In contrast with Madgwick's original work, the above written functions are an enhancement to reduce the equation and decouple magnetic and gravity vectors by enforcing an orthogonalization criteria between both vectors. They must be perpendicular. This procedure was also proposed by Wilson et al. in [80]. In contrast to their implementation, this work employs a positive x-axis unit vector as the reference direction for the magnetic field vector instead of a negative y-axis vector. Using x-axis rather than the y-axis definition satisfies the North-East-Down (NED) and right-handed coordinate frame convention. In this convention, the north direction coincidences with the x-axis and the z-axis coincidences with the down direction, compare figure 5.5a for a depiction of the NED convention. Therefore, the objective functions are adapted to suit this.

The combination of gravity and magnetic field measurements enables a complete orientation to be computed. Usually, a satisfying solution calculated using a gradient descent approach would require multiple iterations to reach its minimum value, but as Madgwick et al. [36] state it is acceptable to compute one orientation per time sample providing that the convergence rate is equal or greater than the physical rate of change of the device. This is governed by the step size ( $\mu_t$ ). Equation (4.18) calculates a complete quaternion based on the previous orientation and solution from the GDA per time sample  $t$

$${}^N_B \mathbf{q}_{\nabla, k+1} = {}^N_B \mathbf{q}_k - \mu_k \frac{\nabla f}{\|\nabla f\|} \quad (4.18)$$

## 4.2 Robust quaternion-based Kalman filter

---

where  ${}^N_B\mathbf{q}_{\nabla,k+1}$  is the quaternion computed via the GDA and  $\mu_t$  can be calculated as follows

$$\mu_k = \alpha \| {}^N_B\dot{\mathbf{q}}_{\omega,k} \| \Delta t. \quad (4.19)$$

The initial and original mathematical derivation of the gradient-descent based filter algorithm can be found in [36].

## 4.2 Robust quaternion-based Kalman filter

It is proposed to fuse the quaternion ( ${}^N_B\mathbf{q}_{\nabla,k+1}$ ) calculated on the basis of accelerometer and magnetic field sensor data and the quaternion ( ${}^N_B\mathbf{q}_{\omega,k+1}$ ) calculated on the basis of gyroscope data by using a linear Kalman filter. The computation of only one quaternion from both the magnetometer and accelerometer data reduces the size of the needed model and state vectors since the only input measurement is the four component unit quaternion. Furthermore, (4.8) and (4.9) enables the filter to switch between the MARG- and the IMU-mode if the sensor is subject to short-time magnetic disturbances, thereby remaining the effectiveness upon implementation of the GDA algorithm. The state vector  $x_{k+1}$  of the proposed Kalman filter consists of the quaternion components

$$x_{k+1} = {}^N_B\mathbf{q} = (q_0 \quad q_1 \quad q_2 \quad q_3)^T. \quad (4.20)$$

Following (4.1) and (4.3) for orientation estimation the predicted quaternion is computed through integration of the angular rate measured by the tri-axis gyroscope [79]. The vector differential equation describing the rate of change (4.1) in matrix form can be rewritten as

$${}^N_B\dot{\mathbf{q}}_{\omega,k} = \Omega({}^B\omega_k) {}^N_B\mathbf{q}_k \quad (4.21)$$

where

$$\Omega({}^B\omega_k) = \begin{pmatrix} 0 & -{}^B\omega_{x,k}^T & -{}^B\omega_{y,k}^T & -{}^B\omega_{z,k}^T \\ {}^B\omega_{x,k} & 0 & {}^B\omega_{z,k} & -{}^B\omega_{y,k} \\ {}^B\omega_{y,k} & -{}^B\omega_{z,k} & 0 & {}^B\omega_{x,k} \\ {}^B\omega_{z,k} & {}^B\omega_{y,k} & -{}^B\omega_{x,k} & 0 \end{pmatrix}. \quad (4.22)$$

Discretization and added process noise ( $w_k$ ) to compensate for model errors results in the discrete state transition equation

$$x_{k+1} = \Phi({}^B\omega, \Delta k)x_k + w_k \quad (4.23)$$

where  $\Phi({}^B\omega, \Delta t)$  denotes the state transition matrix and is computed using zero order integration yielding

$$\Phi({}^B\omega, \Delta t) = \left( I_{4 \times 4} + \frac{1}{2} \Omega({}^B\omega) \Delta t \right). \quad (4.24)$$

The discrete process noise ( $w_k$ ) can be modelled according to (4.1), as the noise affecting the gyroscopes reading as a quaternion derivative, where instead of  $\omega$  white Gaussian measurement noise  $v_k$  with covariance matrix  $\Sigma_g$  is being used

$$\Sigma_g = \sigma_g^2 I_{3 \times 3} \quad (4.25)$$

where  $\sigma_g$  is the standard deviation of the gyroscope readings.

Switching multiplicands and expanding (4.1)  $w_k$  can be written as

$$w_k = -\frac{\Delta t}{2} \begin{pmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & -q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{pmatrix} v_k. \quad (4.26)$$

According to (4.26) the process noise covariance matrix  $Q_k$  can be computed

$$Q_k = \left(\frac{\Delta t}{2}\right)^2 \begin{pmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & -q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{pmatrix} \Sigma_g \begin{pmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & -q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{pmatrix}^T. \quad (4.27)$$

The update step applies the quaternion  ${}^N_B \mathbf{q}_{\nabla,k}$  from (4.18) and adds measurement noise. The discrete process model equation for the update can simply be written as:

$$z_k = {}^N_B \mathbf{q}_{\nabla,k} = Hx_k + v_{zk} \quad (4.28)$$

where  ${}^N_B \mathbf{q}_{\nabla,k}$  represents the quaternion computed through GDA,  $H$  is a  $4 \times 4$  identity matrix and  $v_{zk}$  is noise associated with the estimation of orientation through the GDA based on the sensor's measurement noise which is assumed to be white. Noting that the quaternion computed through GDA is based on accelerometer and magnetometer, thus  $v_{zk}$  is related to accelerometer and magnetometer measurement noise.

The measurement noise covariance matrix ( $R_k$ ) takes a major role regarding the fusion process of the filter. Under dynamic motion the accelerometer will be subject to dynamic linear acceleration which influences the measured direction of the gravity vector thus resulting in a wrong computation of orientation inside the GDA. To avoid the effect of a wrong measurement quaternion onto the orientation estimation, a gain factor is introduced to weight the fusing of the measurement quaternion with the quaternion computed via angular velocity. In case of a Kalman filter this fusion gain (or trust) is implemented in the measurement noise covariance matrix. In phases of high dynamic motion, the covariance matrix should be updated with a high covariance since the measurement uncertainty rises. The magnitude of the covariance gain scales with respect to the physical rate of change is resulting in a decreasing influence of the measurement on

the solution of the Kalman filter, since the gyroscopes noise variance is significantly lower during these motions. In contrast, the variance of the accelerometer and magnetometer measurement in static or slow movement phases is smaller. This is due to the small linear acceleration adding up onto the gravity estimate, which in turn is used to horizontalize the magnetometer measurements. During these phases, the contribution of the measurement quaternion should be high, keeping gyroscope errors at a minimum [41]. To model this noise variance distributions, the initially measured measurement covariance matrix is multiplied by the step size  $\mu_k$  of the GDA. In phases of high dynamic motion, the step size will rise due to increased angular rates resulting in a solution heavily favouring the gyroscope measurement. At rest or slower movements, the orientation estimation is mainly composed of the measurement quaternion from GDA.  $R_k$  can be written as

$$R_k = \Sigma_{a,m} + \mu_k, \Sigma_{a,m} = E[v_{zk}v_{zk}^T] \quad (4.29)$$

where  $\Sigma_{a,m}$  represents the covariance matrix and equals noise introduced from both sensors affecting the estimation of a quaternion inside the GDA. It is determined using measurement data and results in a diagonal  $4 \times 4$  matrix. Since the process of generating the measurement quaternion through gradient descent algorithm is based on measurement data of the accelerometer as well as magnetometer, the process noise is related mainly to the corresponding noise of the sensors to compute the measurement quaternion [61] [81]. The GDA converges over time, as long as the physical rate of change is smaller than the convergence rate which is governed by the step size  $\mu_k$  [36]. Therefore, the step size delivers the main contribution to the covariance change during varying motion dynamics.

### 4.3 Maintaining orientation under magnetic disturbance

It is possible to calculate orientation either through MARG or IMU like filter fusion by switching in between equation (4.8) and (4.9) in the update step of the filter. Using both the GDA and the Kalman filter it is possible to detect magnetic disturbance or uncalibrated magnetometer measurements. This is crucial because in distorted fields the quaternion calculated from GDA will not result in the correct orientation.

Within this work, the following concept is proposed: The calculation of the full MARG gradient is separated from the gyroscope readings. The estimated quaternion from magnetometer and accelerometer data in equations (4.9) and (4.18) is used as an input to the Kalman filter. The quaternions from the MARG- and IMU-equations are used as measurement input to the linear Kalman filter. This reduces the time needed to converge to a steady state inside the GDA and allows to detect magnetic disturbance inside the Kalman filter. The block diagram in figure 4.3 depicts the concept. The quaternion calculated on the basis of acceleration data and magnetic field data represents the input  $z_{\nabla,k}$ . If the innovation ( $y_k$ ), the difference between the measurement quaternion from the GDA

in the MARG-case and the quaternion computed using the gyroscope data, significantly deviates in the update step, it can be assumed, that magnetic disturbances are present

$$y_k = z_{\nabla,K} - Hx_k. \quad (4.30)$$

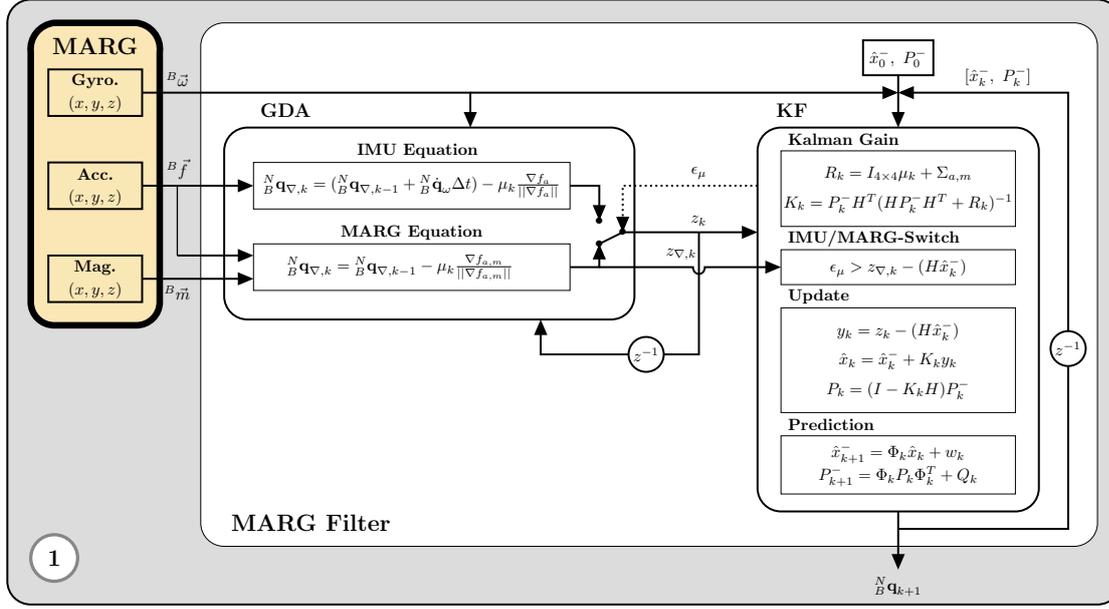
To avoid the effect of short time disturbed magnetic fields on the quaternion and therefore the estimated yaw angle, it is possible to switch the measurement quaternion equation ( $z_k$ ) to the IMU case of equations (4.8) through threshold-based switching. Therefore  $|y_k|$  is computed and tested for large differences against a criteria threshold  $\epsilon_m$ . This constant is the criteria for magnetic disturbance recognition and is determined through experiments to provide best switching capabilities while subject to magnetic disturbance. Upon switching towards the IMU equation, the GDA solution does miss a heading reference. The measurement quaternion would therefore only contain pitch and roll corrections while zeroing out the heading component because the quaternion computed on basis of the accelerometer only defines the vertical z-axis. Since the quaternion is directly used as the measurement input, the filter would converge towards a zero heading orientation, effectively introducing a large artificial error. To account for this, it is necessary to adapt the GDA input quaternion ( ${}^N_B \mathbf{q}_{\nabla,k}$ ) to incorporate the gyroscope readings to account for the missing heading information. During this thesis, two possible options have been tested and verified to work and incorporate heading information. The first and published approach [1] substitutes the quaternion  ${}^N_B \mathbf{q}_{\nabla,k}$  with the Kalman filters output quaternion  ${}^N_B \mathbf{q}_{k+1}$  – the a priori state vector  $\hat{x}_{k+1}$ . This formulation would need a high gain factor  $\alpha$  and convergence constant  $c_b$  to ensure a large step-size and big contribution of the gradient calculation on the quaternion. This results in the following equations

$$z_k = \begin{cases} {}^N_B \mathbf{q}_{\nabla,k+1} = {}^N_B \mathbf{q}_{\nabla,k} - (\mu_k + c_b) \frac{\nabla f_{a,m}}{\|\nabla f_{a,m}\|}, & |y_k| < \epsilon_\mu \\ {}^N_B \mathbf{q}_{\nabla,k+1} = \hat{x}_{k+1} - (\mu_k + c_b) \frac{\nabla f_a}{\|\nabla f_a\|} \end{cases}, \quad (4.31)$$

where  $c_b$  is set to 0.1.

In contrast to the original computation formula, the IMU mode equation is slightly modified within the progress of the thesis. The equation is adapted to gain a faster convergence of the gradient solution and decouple it from the Kalman filters a priori state vector as a substitute quaternion.

The heading information is provided using the angular rate measurements to propagate the heading component of the quaternion  ${}^N_B \mathbf{q}_{\nabla,k+1}$  which is achieved using the numerical integration of angular rate in quaternion representation from equation 4.3. In order to provide a fast convergence of the GDA in either static or dynamic conditions and to create a larger difference  $y_k$  under magnetic disturbance, a gain factor  $c_b$  is added to the



**Figure 4.3:** Block diagram of the developed core filter framework incorporating a GDA-based filter stage to compute a complete measurement quaternion into a linear Kalman filter to enable robust orientation estimation under short-time magnetic disturbances.

calculation of (4.8) and (4.9) resulting in the following cases

$$z_k = \begin{cases} \frac{N}{B} \mathbf{q}_{\nabla, k+1} = \frac{N}{B} \mathbf{q}_{\nabla, k} - (\mu_k + c_b) \frac{\nabla f_{a,m}}{\|\nabla f_{a,m}\|}, & |y_k| < \epsilon_\mu \\ \frac{N}{B} \mathbf{q}_{\nabla, k+1} = \left( \frac{N}{B} \mathbf{q}_{\nabla, k} + \frac{N}{B} \dot{\mathbf{q}}_{\omega} \Delta t \right) - (\mu_k + c_b) \frac{\nabla f_a}{\|\nabla f_a\|} \end{cases} \quad (4.32)$$

where  $c_b$  is equal to 0.001. This formulation does increase the complete computational cost by only a single addition, since  $\frac{N}{B} \dot{\mathbf{q}}_{\omega}$  is essentially calculated as part of equation (4.33).

During magnetic disturbance both equations from (4.32) need to be calculated to keep the magnetic disturbance tracked in the form of  $y_k$ , which relies on the MARG-equation ( $z_{\nabla, k}$ ). If the disturbance vanishes the fusion again applies the MARG-case. However, this depends on the period of the disturbance. The IMU based solution is subject to gyroscope drift and therefore  $y_k$  will scale over time making it difficult to get back to the MARG-case without large errors.

The fact that this filter computes a complete quaternion solution introduces one general problem. While an orientation can be expressed as different quaternions the filter needs to calculate similar quaternions to switch from IMU to MARG case by comparing the GDA quaternion and the state vector. When the sensor is calibrated properly,  $y_k$  will be small. If the sensor is not calibrated,  $y_k$  scales up. A complete rotation of the device around a major axis might end up in different quaternions for the GDA quaternion and

state vector representing nearly the same orientation. In such cases,  $|y_k|$  will become either 2 or 0 because of the norm of the subtraction of two unit-quaternions. At this point the orientation is the same but the quaternions differ. This case can be addressed by setting the previous state vector  $x_k$  to  $z_k$  resolving different quaternions for the same orientation representation.

Finally, the step size  $\mu_k$  can be computed. The adaptive step size is based on the rate of change of orientation. This rate of change can be expressed as the quaternion from a previous time step multiplied by half of the angular velocity vector expressed as quaternion. The quaternion from the previous time step is set to be the measurement quaternion calculated through the gradient descent algorithm. This is done so that  $\mu_k$  will scale up while under dynamic motion and to consider different sampling times. Usually, accelerometer and gyroscopes can be sampled at much higher rates than a magnetometer. During dynamic motion the GDA will therefore have a larger step size resulting in a fast convergence, even when it is limited through magnetometer sampling time until the motion becomes static or slow. Equation (4.19) expects the rate of change of the orientation to compute an adaptive step size. This rate of change can be computed through (4.1), where  $\frac{N}{B}\mathbf{q}_k$  is formed through (4.32) which yields that  $\mu_t$  can be calculated as follows

$$\mu_t = \alpha \left\| \left( -\frac{1}{2} \begin{pmatrix} 0 \\ \vec{\omega}_b^n \end{pmatrix} \bullet z_{k-1} \right) \right\| \Delta t. \quad (4.33)$$

A complete diagram of both filter stages and their dependencies is depicted in figure 4.3.

## 4.4 Setup and experimental results

The initially proposed filter was implemented in MATLAB and tested using an FSM-9 module from Hillcrest Labs. The FSM-9 consists of a tri-axis gyroscope, magnetometer and accelerometer and features full 3D motion processing on a 32-bit MPU, including quaternion output<sup>5</sup>. The FSM-9 was programmed to output factory-calibrated values sampled at a rate of 125 Hz, and directly feed to the proposed filter. The sensor was calibrated to the surrounding magnetic field according to the manufacturer's advice.

A robot (UR5) was used to rotate the sensor to provide accurate orientation changes that are used as the ground truth to compare the algorithms. The motor encoders of the UR5 have a resolution of  $0.01^\circ s^{-1}$ . A second MARG sensor, the XDK110 Cross-Domain Development Kit from BOSCH, was programmed to output raw values, calibrated according to [48] working as a redundant source of reference to validate correctness of data. A set of 16 rotations around every major axis ( $x$ ,  $y$ ,  $z$ ) of the sensor, where each rotation corresponds to  $180^\circ$  total orientation change in the particular axis was recorded. Regarding dynamic and static performance of the filter each rotation sequence was recorded with decreasing angular rates, starting at an angular rate of  $18^\circ s^{-1}$  on the first rotation and

<sup>5</sup><http://hillcrestlabs.com/products/fsm-9/>

#### 4.4 Setup and experimental results

**Table 4.1:** RMS values for commercial implemented filter (FSM-9) and proposed filter (KFGDA) under three different movement constraints – static, slow dynamic and fast dynamic.

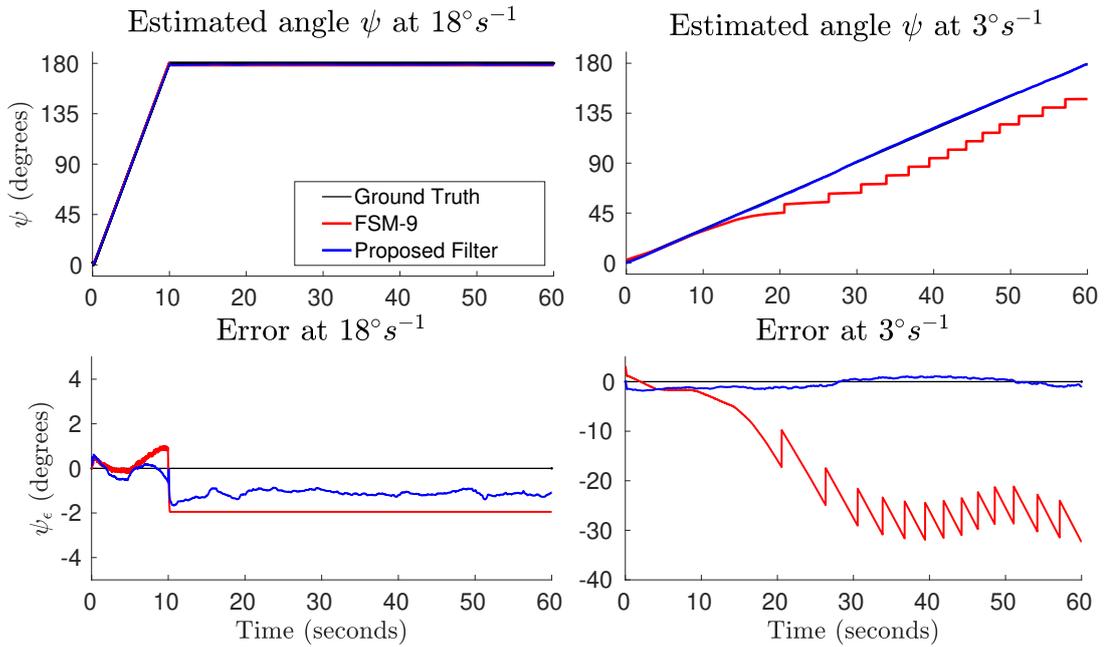
Angle error	Static	Dynamic (slow)	Dynamic (fast)
	FSM-9   KFGDA	FSM-9   KFGDA	FSM-9   KFGDA
$\phi_\epsilon$	0.55°   0.54°	0.51°   0.71°	0.51°   0.78°
$\theta_\epsilon$	0.48°   0.45°	0.41°   0.62°	0.33°   0.41°
$\psi_\epsilon$	1.31°   1.23°	24.55°   1.25°	1.27°   1.12°

completing at  $3^\circ s^{-1}$ . Every set was recorded five times for the undistorted case and five times for the magnetically distorted case to investigate behavior under magnetic disturbance. Artificial magnetic distortions were introduced by bringing an iron bar close to the sensors (4 cm) for less than 35 seconds disturbing the surrounding magnetic field by introducing soft iron effects.

The performance of orientation estimation is presented as static and dynamic RMS errors (subscript  $\epsilon$ ) of the Euler angles, roll ( $\phi$ ), pitch ( $\theta$ ) and yaw ( $\psi$ ). These are calculated as the difference between estimated orientation and the ground truth obtained from the robot joint angle. The proposed algorithm parameters for the initial test setup are as follows:

- Parameters of noise:  $\sigma_g^2 = 0.00003136$ ,  $\Sigma_{a,m} = 0.01$
- Initial states: State covariance matrix,  $P_0$ , was initially set to be the identity matrix  $I_{4 \times 4}$  and the state vector,  $x_0$ , was set to unit quaternion  $[1 \ 0 \ 0 \ 0]^T$ .
- Gradient-descent scaling factor of the initial implementation is  $\alpha = 10$ . The magnetic disturbance threshold  $\epsilon_\mu = 0.065$ .
- At startup,  $c_b$  was set to 5 and  $\epsilon_\mu$  to 50 for 5 seconds to let the GDA converge faster towards the initial orientation defined by direction of gravity and magnetic field vector. The gradient descent algorithm will start from the given orientation  $(1, 0, 0, 0)$  and will converge towards the absolute initial orientation from the sensor defined by the reference vector measurements. Setting  $c_b$  and  $\epsilon_\mu$  values higher, the algorithm will converge much faster and this also provides, that this orientation change will not be interpreted as magnetic disturbance due to the large error between measurement and Kalman filter prediction at startup.

The estimated angle errors are divided into three groups, static, slow, and fast dynamic – where static equals no motion, slow dynamic equals angular rates of  $3^\circ s^{-1} \leq \omega \leq 6^\circ s^{-1}$  and fast dynamic is  $\omega > 6^\circ s^{-1}$ . At static and fast dynamic motions both the commercial



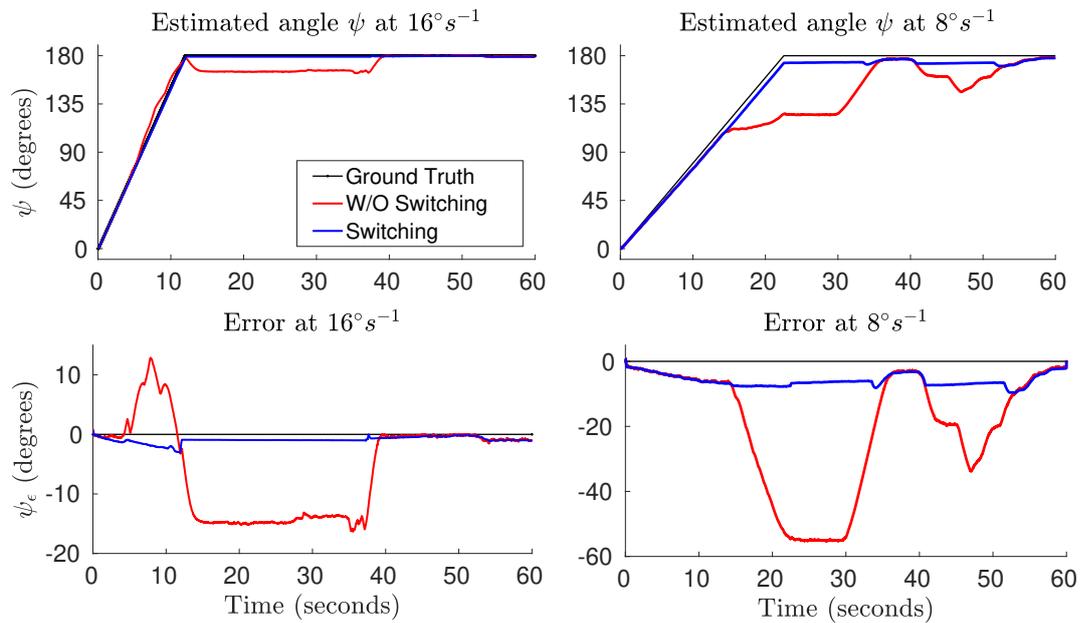
**Figure 4.4:** Typical results for ground truth and estimated angle  $\psi$  (top) for the proposed (blue) and the commercial data fusion (red) algorithms and the corresponding error (bottom) at fast ( $18^\circ s^{-1}$ ) and slow ( $3^\circ s^{-1}$ ) movements.

filter and the one proposed here show similar performance regarding motion estimation (figure 4.4, left). In the case of slow dynamic motion, the commercially available sensor fusion algorithm of the FSM-9 module provides stepwise increasing orientation data with large value steps whereas the proposed filter calculates increasing orientation data narrow to the ground truth as can be seen in figure 4.4 (right) for  $\omega = 3^\circ s^{-1}$ . Table 4.1 summarizes the results for all test sets subdivided into the three categories of motion. The proposed filter can withstand with commercially available devices and is superior at slow dynamic motion regarding orientation estimation. Even at low angular rates of  $3^\circ s^{-1}$  the proposed filter is still able to provide good estimation results with the same sensor data as the commercial algorithm.

Furthermore, figure 4.5 indicates that the proposed algorithm is able to maintain orientation under magnetic disturbance. Typical results of orientation estimation with and without switching between MARG and IMU equations are presented. For fast dynamic motion at  $16^\circ s^{-1}$  the disturbance is introduced 5 seconds after the start of the set and kept for an interval of 35 seconds. At  $8^\circ s^{-1}$  the disturbance is introduced twice, at 15 seconds after start for a length of 20 seconds and at 40 seconds after start for a length of 15 seconds. Switching between MARG and IMU equations inside the GDA enables to maintain orientation in static cases as well as for dynamic motion while magnetic disturbance is present. In contrast, the estimated yaw angle will experience large errors without the proposed switching.

However, in the case of magnetic disturbance lasting more than 35 seconds the orienta-

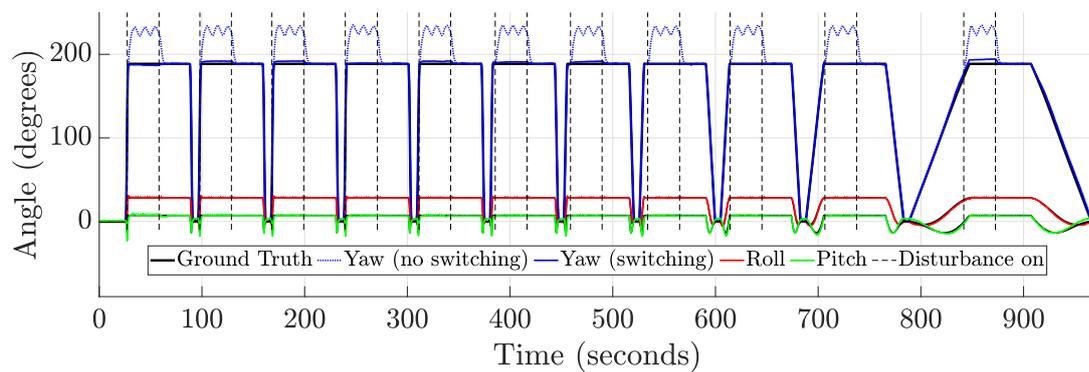
#### 4.4 Setup and experimental results



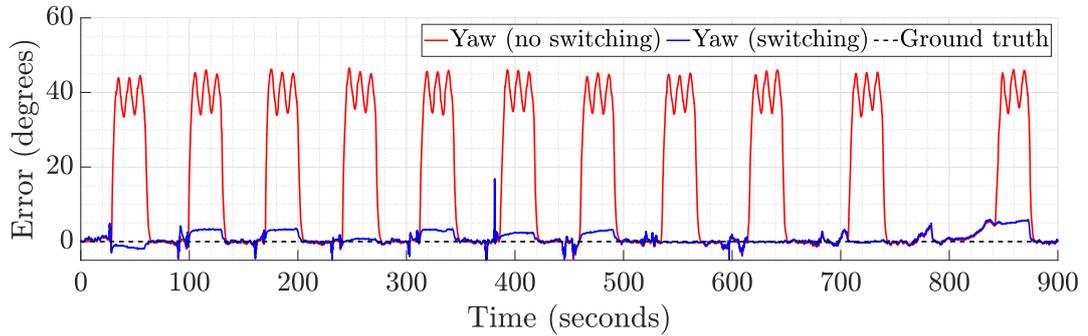
**Figure 4.5:** Typical results for short time magnetic disturbance rejection between ground truth and estimated angle  $\psi$  (top) with (blue) and without (red) disturbance rejection and corresponding error (bottom) at fast ( $16^\circ s^{-1}$ ) and slow ( $8^\circ s^{-1}$ ) movements.

tion estimation will experience errors in the heading estimate with respect to time. This is due to the gyroscope bias and gyroscope resolution making an orientation estimation without large correction steps and growing errors impossible.

The slightly adapted algorithm was tested using the WiOT system with a broader set



**Figure 4.6:** Results for ground truth and estimated Euler angles (yaw, roll and pitch) for various motion dynamics. The dashed black line indicates magnetic disturbance, the dotted blue line depicts the yaw estimate while subject to magnetic disturbance (no compensation) while the solid blue line depicts results with short-term magnetic compensation.



**Figure 4.7:** Error graph of magnetic disturbance rejection between ground truth (black) and estimated yaw angles at different rotation velocities – with short-time magnetic disturbance rejection (blue) and without rejection (red).

of rotational velocities ranging from  $180^\circ s^{-1}$  to  $3^\circ s^{-1}$  with  $20^\circ s^{-1}$  decrease of rotation velocities between the rotations resulting in a total set of 11 rotations. In contrast to the first experimental setup the sensor is simultaneously rotated in all three axis –  $180^\circ$  around the z-axis,  $30^\circ$  around the x-axis and  $10^\circ$  around the y-axis – to validate the filters performance during coupled dynamic motions. Furthermore, a second robot arm is used to introduce the disturbance (iron rod). The iron rod is brought close to the sensor (25 cm) by the second robotic arm. The robotic arm then moves the iron rod further away (approx. 5 cm) before bringing it back to the initial pose (25 cm) to simulate dynamic disturbances.

The filter’s set of parameters stays the same except for  $\alpha$ , which is set to 0.3 and  $c_b$  which is 0.0003. Figure 4.6 depicts a complete trial data for all three Euler angles. The filter is capable of delivering accurate orientation estimation throughout all rotation velocities without large errors (compare figure 4.7). Maximum error during non disturbed motion results in around  $5^\circ$  error compared to ground truth. The pitch and roll components of the filter are not influence by the influence of the magnetic disturbance, which indicates the benefit of the adapted GDA formulation. Furthermore, it can be seen that the filter is capable of recognising and rejecting short-term magnetic disturbances by switching towards the IMU-mode effectively reducing the error to a maximum of  $5^\circ$  while no magnetic heading reference is available. As soon as the disturbance vanishes, the filter switches back to full MARG-mode and reduces the error.

## 4.5 Conclusion

Cascaded two-step Kalman filter approaches to estimate orientation of MARG sensors and overcome short-time magnetic disturbances are reported since 2014. These filters are based on large models and equation sets which significantly increase complexity and the need for optimal parameter tuning. The filter approach presented here reduces the Kalman filter equations to the gyroscope only, therefore reducing model size

and complexity to the well-known gyroscope dynamics. The calculation of the quaternion from gravity and magnetic field is based on the computationally efficient GDA. Under magnetic disturbance the proposed filter enables switching between MARG and IMU mode. This allows to maintain orientation and therefore increases robustness in environments which are exposed to short-term magnetic disturbance. Separating the gradient-descent quaternion from the gyroscope enables a motion based orientation estimation to be computed thus prioritizing the strengths and weaknesses of the different sensors and allowing a continuous check for magnetic disturbance. Using a Kalman filter improves performance regarding gyroscope errors in phases of magnetic disturbance keeping drifts relatively low. In contrast to Madgwick's original approach, the magnetometer does not necessarily need to be calibrated. Without valid magnetometer calibration, the filter will act as an IMU, switching to the equivalent subset of equations, whereas a valid magnetometer calibration will enhance the performance and precision of the orientation estimation.

This approach nevertheless still has a bottleneck if the magnetic disturbance is persistent or experiences rapid and ongoing changes. During long term perturbations of more than 35 s, the heading estimate will experience drift due to the inevitable accumulation of angular random walk errors or temperature induced changes in the overall noise spectrum of the gyroscope. To account for this kind of error and guarantee a safe human-robot collaboration in the long run, additional measures have to be installed to keep a good attitude. This can be either achieved using additional hardware, functioning as a heading reference directly correcting for accumulated errors, or by exploiting motion constrains, e.g. through ZRU, that can be enforced to keep the gyroscope drift errors low. Additional references usually contribute to higher accuracy and less noise introduced error accumulations, but come at the cost of extra hardware, new measurement values, additional digital filter processing, increasing computational load and sometimes completely different measurement models. Furthermore, the additional references should neither be too sensitive to the environmental conditions nor require any external hardware setup, such that the interface is still usable outside laboratory environments and conditions. It should not restrict the human's motion range or working area, an additional reference therefore needs to meet the infrastructureless criteria. Additional signal processing techniques or motion constraints can be implemented on the sensor board itself and therefore fulfil the infrastructureless requirement but come at the cost of a less accurate error compensation. Furthermore, the constrains might not satisfy the actual use case, e.g. ZRU. If the head is in constant motion, even at very slow rates, the ZRU cannot be utilized. To conclude, to keep a good orientation estimation, further methods, references, and algorithms need to be researched and applied which should ideally not be restricted to the environmental conditions and comply with the infrastructureless criteria and as such should be head-worn.

A possible infrastructureless solution and enhancement to this problem is explored in

the next chapter. It examines the usage of eye tracking data, specifically pupil position measurements, as a potential source to support MARG sensor-based heading estimation. It furthermore utilizes the changed formulation of the magnetometer objective function and Jacobian to use a substitute reference vector if the magnetometer data is not reliable to reduce the needed set of equations and give a new measure for magnetic disturbance detection.

## 5 Eye tracking-supported MARG sensor orientation estimation

The content and results of the following chapter were published in [4] and are extended by further details.

MARG sensors are exposed to magnetic field disturbance, termed soft- or hard-iron effects, resulting in incorrect orientation estimation [82]. This error scales with respect to the distance between the sensor and source of magnetic disturbance and magnetic properties of the source, for example, 35–50° error near large ferromagnetic metal objects or the floor (indoor) [33]. Recent approaches try to overcome these magnetic disturbances with the use of software, for example, switching to IMU-based estimation only, online gyroscope bias estimation [83] or fast online magnetometer calibration [84].

These software-based corrections do not require additional hardware nor other sources of reference but might require certain motion conditions. IMU-based only estimation methods, for example, will experience inevitable accumulation of errors with respect to time, due to the intrinsic noise characteristics of the gyroscope and the fact that no other source of reference is used to correct for it. Online magnetometer calibration approaches are usually based on a sampling of sparse 3D magnetometer data points to adapt the calibration matrix, which might not be possible in every situation due to

A) 3D Head Pose

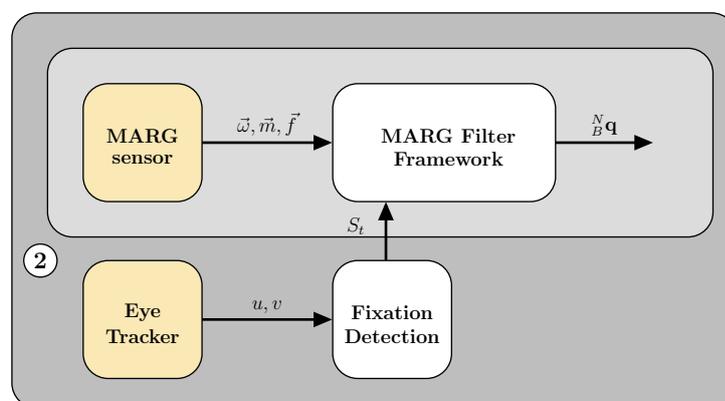
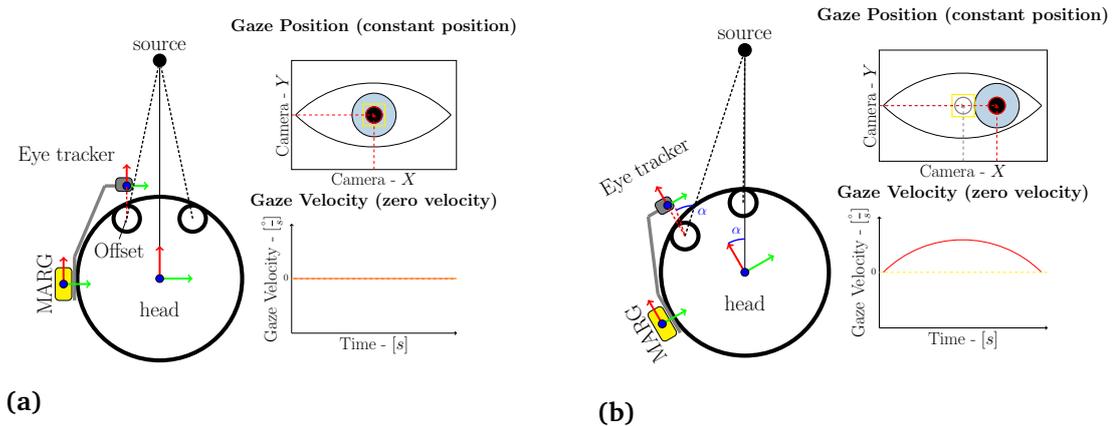


Figure 5.1: Simplified block diagram of the SteadEYE-head data fusion stage.

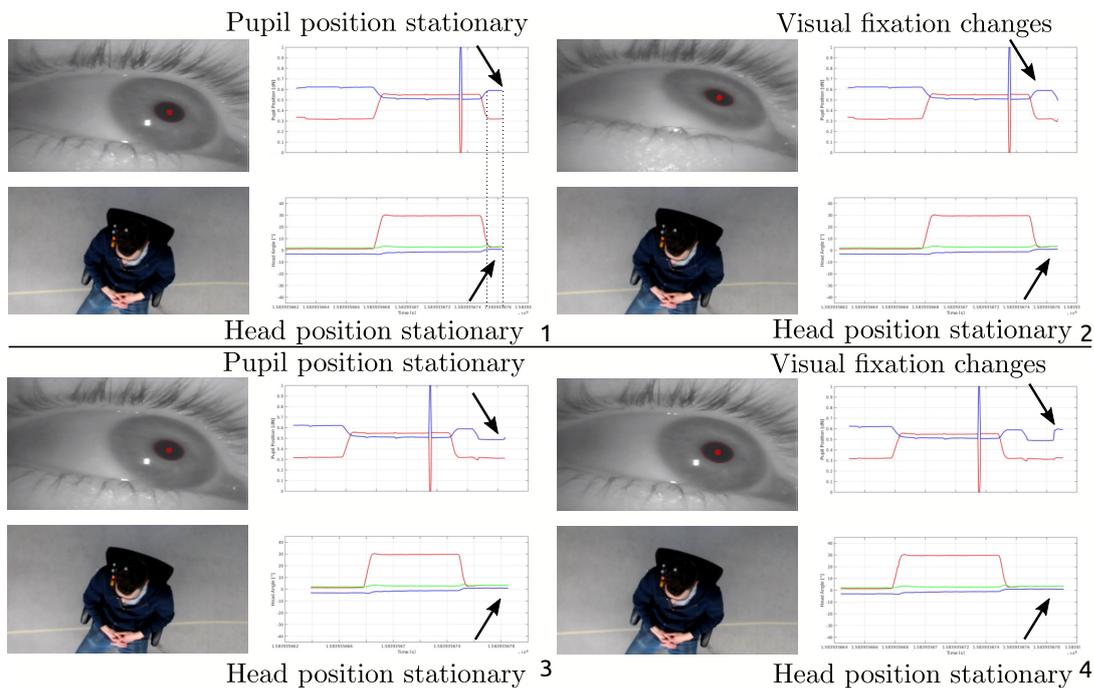


**Figure 5.2:** Coordinate systems between eye tracker, MARG sensor and head when fixating an object (a) without and (b) with head motion. The condition of fixation of the object results in a stable gaze position. Possible motion of the pupil during fixations itself, so-called microsaccades, is very small ( $0.2^\circ$  at a duration of  $20 - 30 [ms]$ ) and can be neglected (yellow boundary box).

fast changing magnetic field values or in this context the rather small motion space of the head. The use of other hardware, for example, visual odometry, usually provides a decent source of reference for the heading estimate but depends on the surrounding environmental conditions, for example, structured environment, reasonable lighting conditions and small relative motion in the scenery [85]. These conditions cannot always be guaranteed, especially in the context of human-machine collaboration which will feature a lot of relative motion from the robotic system and heavy dynamic magnetic disturbances due to the robot's metal-links and motor-joints.

To account for these errors, a novel approach is researched, that reduces the heading estimation errors of head movement measurements by the functional combination of mobile eye tracking and a head worn MARG sensor. The approach utilizes the physiological connection between eye and head movements to identify static and dynamic motion phases. The eye tracking glasses are used to track visual fixations which indicate static phases. This indication is used for zero-orientation change updates in a MARG sensor data fusion algorithm. The approach relies on an infrared based eye camera only and does not need a scene or scene camera and therefore no eye to scene camera calibration. The presented approach is decoupled from most surrounding environmental conditions, and therefore infrastructureless, see figure 5.1 for a simplified block diagram.

The working principle is based on the physiological relationship between eye- and head-rotations during visual fixations of stationary objects. The eyes of a human are centered in a fixed axis of rotation inside the head and are therefore naturally affected by head rotations. Visual fixations of objects will result in small or non-significant rotations of the eyeball during stationary motion phases, see figure 5.2a. A rotation of the head dur-



**Figure 5.3:** Pupil position change due to fixation switch of the eyeball. The head does not experience any orientation change. Visual fixation changes are coupled with fast and abrupt changes in positions.

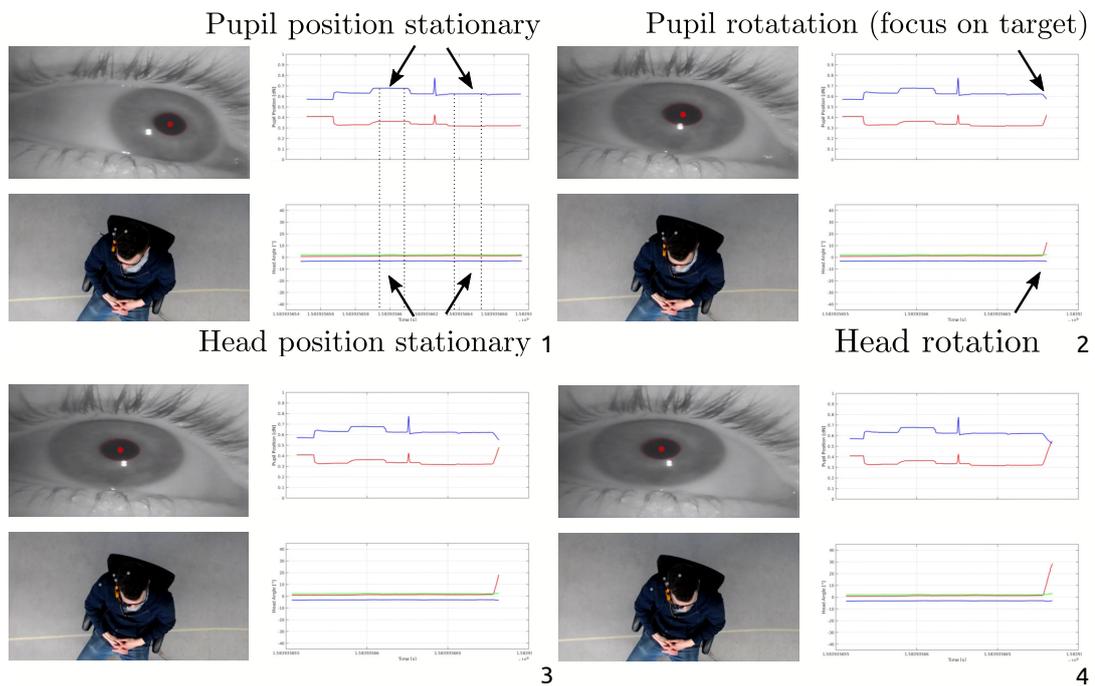
ing visual fixation of a stationary object, however, will result in an opposite rotation of the eyeball due to the Vestibulo-Ocular Reflex (VOR), stabilizing the visual scenery [53], see figure 5.2b. The physiological relationship between head- and eye-rotations therefore represents a natural indicator for head rotation and can be used to support head-orientation measurements.

A mobile eye tracker is used to measure the above-mentioned relationship and utilize this indicator for MARG sensor-based head-orientation measurements. The method assumes that the MARG sensor is fixed in position with respect to the eye tracker frame, for example, rigidly attached to it. The mobile eye tracker is worn by the user and should be adjusted in a way that prevents heavy slippage during head motion, for example, through an eyewear strap which is common practice in mobile eye tracking. From the setup given in figure 5.2, the following constraints can be derived.

The coordinate system of the mobile eye tracker and MARG sensor share a common reference frame with the user's head and rotate conjointly, see figure 5.2. Every rotation of the head is directly coupled with the rotation of the eye tracker frame and MARG or IMU sensor. This rotation will result in a change of the pupil position in the eye camera image. This is either due to a voluntary change in the visual fixation (compare figure 5.3) or a head movement (compare 5.4). In contrast to head rotation based changes of the pupil position, a change of visual fixation is usually coupled with high angular velocity of the eyeball due to a significant pupil position change of consecutive eye camera images.

If the visual target is moving while the head is stationary or the fixation changes to another visual target, there will be a significant change in pupil position, due to the given coordinate system setup. The eye tracker camera is in a fixed position with respect to the head coordinate system. If the eyes follow a moving target or switch the visual fixation, the pupil position changes within consecutive frames because the target changes its position with respect to the head- and eye tracker coordinate system.

Changes in the pupil position will therefore always indicate motion, whether it is introduced through head motion or voluntary eye motion. Near-zero changes in the pupil position between eye tracker camera images, however, indicate near-zero head rotation with one possible exception from this assumption. If a visual fixation stays on a moving target while the head is rotating at the same rotational speed at which the target is moving, all local coordinate systems do keep their relative positions between each other. This would result in a no motion classification from the pupil position change criteria. In this kind of situation, the pupil position does not change with respect to the eye tracker coordinate system, since the head- and target-coordinate system do not change their relative positions and orientations between each other. A MARG or IMU sensor, however, does measure motion related to the world coordinate system and will there-



**Figure 5.4:** Stationary and synchronous head- and eye-ball rotation sequence. The dashed lines indicate phases of zero pupil-position change and zero head-orientation changes (image 1). If the eyes stay on the visual area of interest and the head starts moving (rotates around the z-axis), the eyes create a counter-rotation to stabilize and fixate the visual point of interest (images 2-4).

fore measure a change in the orientation between the world- and eye tracker coordinate system. The rotational velocity of the motion, or in other words the change of orientation between coordinate systems, needs to exceed a minimum threshold to distinguish the motion from gyroscope noise during these special phases.

The approach presented here uses this pupil motion description under visual fixation of an object to reduce the drift effect in stationary phases and therefore improve the MARG sensor-based heading estimate. Since all local coordinate systems are in a fixed position towards each other, every simultaneous measured movement or motion is caused by head rotations. Every significant pupil position change indicates motion, either from head rotation or eye rotation. Zero or no rotation, however, is indicated by every visual fixation, independent of the total fixation time that results in a near-zero change of the pupil position and angular velocity.

### 5.1 SteadEYE-head: Using eye tracking data to improve heading estimates

In this work it is proposed to support MARG sensor based heading estimates by zero-rotation updates measured and indicated by visual fixation detection. The detection of visual fixation (given in subsection 5.1.1) is used to feed the previous estimate of heading from the MARG sensor fusion process (given in subsection 5.1.2) recursively to the filter itself to reduce accumulation of gyroscope ARW and DC bias-related heading errors. An IMU heading vector  ${}^B\vec{N}_{\text{IMU},k}$  is calculated based on the previous estimate of the heading that represents the direction of a horizontalized heading vector in the North-East-Down (NED) frame. This heading vector can be used as a complete substitute to the magnetometer-based horizontalized north direction vector in the MARG-equation of the adapted form of Madgwick's gradient-descent filter in the Kalman filter framework in this work. The method, however, can be implemented and used within any filter approach and hardware combination.

Synchronization of both systems is achieved through timestamp-based message filtering. The data of the mobile eye tracker as well as the used MARG or IMU sensor should be accessible in real time by the manufacturer's API and provide a timestamp that can be used for synchronization processes, for example, using the message filter package from the robot operating system (ROS) framework. An angular rate threshold-based switching can be either implemented on the MARG sensor or host computer to account for possible latency issues between both systems. This threshold is based on the median head motion noise in static motion phases indicated by visual fixations. If the gyroscope raw signal exceeds the median noise level, the zero-orientation update is turned off. This median noise threshold is also used to address the special case that the pupil position does not change while the head and eyes are following a moving target at the same rota-

tional speed. During these special motions the magnitude of the measured angular rate will exceed the median gyroscope threshold which in turn disables the zero-orientation update.

### 5.1.1 Visual zero-rotation detection

The trigger signal for the zero-rotation update is based on an online visual fixation detection algorithm that utilizes dispersion (spatial movement,  $th_d$ ) and duration ( $th_t$ ) thresholds to identify fixations. These thresholds define the total amount of allowed gaze or pupil position differences ( $\Delta p$ ) between time successive eye camera images ( $\Delta t$ ). The algorithm utilizes a sliding window which length is determined by the duration threshold  $th_t$  and sampling frequency. Dispersion  $p$  is calculated as the sum of the differences between consecutive pupil positions [86]

$$\Delta p = [(max(x) - min(x) + (max(y) - min(y))), \quad (5.1)$$

where  $x$  and  $y$  are the eye tracker camera's pixel positions. The dispersion is compared to the maximum dispersion threshold  $th_d$ . Fixations are identified as such if the dispersion stays below  $th_d$ . This results in an expansion of the window to the right until the dispersion exceeds this threshold. If no fixation is detected, the window does not expand but moves forward in time [86].

This kind of algorithm has proven to be very accurate and robust regarding online fixation identification but needs careful parameter setting [86]. While the visual fixation stays on a target and inside the dispersion threshold boundaries, the head is assumed to be stationary. The threshold parameter ratings for the magnitude of dispersion in time is given due to involuntary movement, for example, microsaccades and tremor. However, these involuntary movements usually consist of rather small duration in the range of 20–30 ms and amplitudes peaking in a visual angle of  $0.2^\circ$  [87]. The fixation detection parameters should be chosen in a way that fixations are still detected even in the presence of microsaccades and tremor. A fixation is identified and labelled as such as soon as the fixation duration threshold is reached. Upon this a trigger signal ( $S_k$ ) is emitted indicating a zero-orientation update cycle for the MARG sensor data fusion process.

$$S_k = \begin{cases} 1, & \Delta p \leq th_d \wedge \Delta t \leq th_t \\ 0.0 & \end{cases} \quad (5.2)$$

The trigger starts an acquisition cycle that stores gyroscope raw data while the fixation holds true. When a sufficient amount of gyroscope samples has been recorded, updated motion noise parameters are sent to the MARG sensor to update the threshold to account for desynchronization and latency issues between both systems and their different sampling rates. This procedure ensures adaptive and individual noise parametrization for the current user and use case and enables a real-time support.

### 5.1.2 MARG sensor data fusion

In general, the approach can be used independently of the underlying MARG sensor data fusion process since it indicates whether the users head is in dynamic or static motion phases. In this work, the approach is exploited on an adaptation version of the developed two-stage Kalman filter from chapter 4. Figure 5.6 depicts the complete filter fusion approach that will be explained in detail in the following subsection. As proposed by Madgwick et al., a quaternion  ${}^N_B\mathbf{q}$  is computed by solving a minimization problem

$$\min_{{}^N_B\mathbf{q} \in \mathbb{R}^4} f({}^N_B\mathbf{q}, {}^N\vec{d}, {}^B\vec{s}), \quad (5.3)$$

that rotates a vector  ${}^N\vec{d}$  into the orientation of a reference vector  ${}^B\vec{s}$

$$f({}^N_B\mathbf{q}, {}^N\vec{d}, {}^B\vec{s}) = {}^N_B\mathbf{q} \bullet \begin{pmatrix} 0 \\ {}^N\vec{d} \end{pmatrix} \bullet {}^N_B\dot{\mathbf{q}} - \begin{pmatrix} 0 \\ {}^B\vec{s} \end{pmatrix}, \quad (5.4)$$

where  ${}^N_B$  denotes the orientation of the global navigation frame relative to the body frame and  ${}^N_B\mathbf{q}$  is the four-component quaternion

$${}^N_B\mathbf{q} = (q_1 \quad q_2 \quad q_3 \quad q_4)^T. \quad (5.5)$$

Recall the gradient-descent algorithm as a solution to the optimization problem by solving for the obtained magnetometer and accelerometer vector measurements

$${}^N_B\mathbf{q}_{k+1} = {}^N_B\mathbf{q}_k - \mu_k \frac{\nabla f({}^N_B\mathbf{q}_k, {}^N\vec{d}, {}^B\vec{s})}{\|\nabla f({}^N_B\mathbf{q}_k, {}^N\vec{d}, {}^B\vec{s})\|}, \quad k = 0, 1, 2, \dots, n, \quad (5.6)$$

where  $\mu_k$  denotes the step size of the gradient function. The presented GDA filter stage from section 4 computes a complete quaternion  ${}^N_B\mathbf{q}_{k+1}$  either based on gyroscope, magnetometer, and accelerometer (MARG-case) or gyroscope and accelerometer only data (IMU-case). This setup enables the reduction of errors in the heading estimate from magnetic disturbances, but requires two different sets of equations [1]. This is due to the missing magnetometer measurement vector in the IMU case set of equations and therefore needs an adapted objective function and Jacobian respectively.

To reduce the needed set of equations and get rid of the necessity to calculate two quaternions in parallel, it is feasible to calculate an IMU heading vector that substitutes the magnetometer vector while magnetic disturbance is present to reduce the needed sets of equations as well as to guarantee convergence and a continuous quaternion solution to the minimization problem. If no magnetic disturbance exists, the filter can be supplied with the north direction vector  ${}^B\vec{N}_m$  from the NED formulation through accelerometer and magnetometer measurements. Recall the process for the north direction vector calculation from section 4, equations (4.10)-(4.12). The north direction vector is defined as the combination of two cross products between the accelerometer

and magnetometer vectors,

$${}^B\vec{N}_m = -{}^B\vec{a} \times \left( -{}^B\vec{a} \times {}^B\vec{m} \right), \quad (5.7)$$

This formulation however, can lead to inaccuracies if the device is in motion. During rotation the acceleration vector will be subject to motion acceleration and therefore does not accurately reflect the direction of gravity. This effect is typically reduced by low-pass filtering the acceleration vector. Most modern sensors provide on-board low-pass filter banks that can be configured by the user according to the appropriate needs. Furthermore, the rotational accelerations will be rather small compared to the dominant acceleration originating from gravity. This is due to the small distance (in this case 0.1 m) between the rotational center of the head and the position of the MARG sensor resulting in minor inaccuracies during dynamic motion. The influence of these inaccuracies during dynamic motion is further reduced by the subsequent data fusion filter. The data fusion inside the linear Kalman filter developed in this thesis does emphasize on gyroscope data integration during fast dynamic motion to reduce inaccuracies from the motion acceleration on the orientation estimation and therefore enhances estimation robustness. This dynamic motion weighting can be used to reduce the effect by computing the down vector using the filters orientation estimation output ( ${}^N_B\mathbf{q}$ ). This is achieved via two-sided quaternion multiplication using equation 5.4 to transform the reference vector  ${}^N d_a$  from equation (4.13) into the estimated direction of acceleration in the body coordinate system, resulting in

$${}^B D = {}^N_B\mathbf{q} \bullet \begin{pmatrix} 0 \\ {}^N d_a \end{pmatrix} \bullet {}^N_B\dot{\mathbf{q}}, \quad (5.8)$$

where  ${}^N d_a = (0 \ 0 \ -1)^T$ . This vector attenuates high linear accelerations components during dynamic motion phases since it is computed from the full filtered orientation estimation that emphasizes gyroscope readings during strong motion phases.

To substitute the north direction vector during disturbances, an IMU heading vector  ${}^B\vec{N}_{\text{IMU}}$  is calculated, based on the orientation estimation from the gyroscope and accelerometer measurements. This is achieved through the following process.

The heading information of the last output quaternion  ${}^N_B\mathbf{q}_k$  is extracted by calculating a three-component vector ( ${}^B\vec{N}_{\text{IMU},k}$ ) describing heading information in the NED frame. First, the heading information (yaw angle,  $\psi_E$ ) from the quaternion  ${}^N_B\mathbf{q}_k$  is converted to Euler angle representation

$$\begin{aligned} a &= (q_{k,1}^2 + q_{k,2}^2 - q_{k,3}^2 - q_{k,4}^2) \\ b &= 2 \cdot (q_{k,2} \cdot q_{k,3} + q_{k,1} \cdot q_{k,4}) \\ \psi_E &= \text{atan2}(b, a). \end{aligned} \quad (5.9)$$

When a zero-rotation update is triggered, the fusion process samples the current output

angle  $\psi_E$  from the last output quaternion  ${}^N_B \mathbf{q}_k$  of the GDA filter stage and holds it while the trigger  $S_k$  is true. The subscript  $E$  indicates that the angle  $\psi$  is not updated if the sample and hold mechanism is activated. If the trigger signal is false, indicating head motion, the fusion process updates the angle  $\psi_E$  with every new output quaternion  ${}^N_B \mathbf{q}_k$ . The heading information (yaw angle,  $\psi_E$ ) from the quaternion  ${}^N_B \mathbf{q}_k$  is converted to a quaternion representing only the yaw rotation ( ${}^N_B \mathbf{q}_{\psi,E}$ ) by deriving it from the corresponding Euler angle representation [88]. A unit quaternion representing heading information ( ${}^N_B \mathbf{q}_{\psi}$ ) is expressed as a rotation  $\psi$  around the z-axis

$$\begin{aligned} \mathbf{q} &= \begin{pmatrix} \cos(\psi/2) & 0 & 0 & \sin(\psi/2) \end{pmatrix}^T, \\ {}^N_B \mathbf{q}_{\psi} &= \frac{\mathbf{q}}{\|\mathbf{q}\|}. \end{aligned} \quad (5.10)$$

The heading quaternion  ${}^N_B \mathbf{q}_{\psi,E}$  can be expressed without trigonometric functions by substituting the corresponding Euler angle equation (5.9) with (5.10) and normalize it, resulting in

$$\begin{aligned} \mathbf{q} &= \left( ((q_{k,1}^2 + q_{k,2}^2 - q_{k,3}^2 - q_{k,4}^2)) \quad 0 \quad 0 \quad (2 \cdot (q_{k,2} \cdot q_{k,3} + q_{k,1} \cdot q_{k,4})) \right)^T, \\ {}^N_B \mathbf{q}_{\psi} &= \frac{\mathbf{q}}{\|\mathbf{q}\|}. \end{aligned} \quad (5.11)$$

To get the half-rotation angle from equation (5.9), a unit quaternion is added and the result is normalized

$$\begin{aligned} \mathbf{q} &= {}^N_B \mathbf{q}_{\psi} + \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}^T, \\ {}^N_B \mathbf{q}_{\psi,E} &= \frac{\mathbf{q}}{\|\mathbf{q}\|}. \end{aligned} \quad (5.12)$$

When a zero-rotation update is triggered, the fusion process samples the current output quaternion  ${}^N_B \mathbf{q}_{\psi,E}$  from the last output quaternion  ${}^N_B \mathbf{q}_k$  of the GDA filter stage and holds it while it is activated. If the trigger signal is deactivated, the fusion process updates the heading quaternion  ${}^N_B \mathbf{q}_{\psi,E}$  with every new output quaternion  ${}^N_B \mathbf{q}_k$ .

Even if the heading direction does not change, because a fixation takes place, the residual components of the quaternion shall still be corrected by the reference gravity vector. To do this, a quaternion ( ${}^N_B \mathbf{q}_{\phi,\theta,k}$ ) is computed, representing the iterative updated roll ( $\phi_k$ ) and pitch ( $\theta_k$ ) angles based on the current quaternion  ${}^N_B \mathbf{q}_k$ . This is achieved by removing the yaw component of the current quaternion  ${}^N_B \mathbf{q}_k$  through conjugate quaternion multiplication. Therefore, a yaw quaternion  ${}^N_B \mathbf{q}_{\psi,k}$  based on the equation (5.11) is calculated and applied to the conjugate of the current quaternion  ${}^N_B \mathbf{q}_k$

$${}^N_B \mathbf{q}_{\phi,\theta,k} = {}^N_B \dot{\mathbf{q}}_{\psi,k} \bullet {}^N_B \mathbf{q}_k, \quad (5.13)$$

where  $\bullet$  indicates quaternion multiplication and  $\dot{\mathbf{q}}$  represents the conjugate quaternion to  $\mathbf{q}$ .

The final quaternion  ${}^N_B \mathbf{q}_{E,k}$  can be computed by combining the heading quaternion

${}^N_B \mathbf{q}_{\psi,E}$  and the iterative updated quaternion representing only roll and pitch  ${}^N_B \mathbf{q}_{\phi,\theta,k}$  through quaternion multiplication,

$${}^N_B \mathbf{q}_{E,k} = {}^N_B \mathbf{q}_{\psi,E} \bullet {}^N_B \mathbf{q}_{\phi,\theta,k}. \quad (5.14)$$

The quaternion  ${}^N_B \mathbf{q}_{E,k}$  now represents a complete orientation expressed as quaternion and combines heading information from the sample and hold mechanism with current updates regarding roll and pitch information from the filter's output quaternion. This solution does not suffer from gimbal lock and can be used as the input quaternion to equation (5.15). The quaternion is now applied to an x-axis unit vector because the north direction vector defines the sensor's body x-axis, resulting in

$$\begin{aligned} \vec{x} &= (1 \ 0 \ 0)^T \\ {}^B \vec{N}_{\text{IMU},k} &= {}^N_B \mathbf{q}_{E,k} \bullet \begin{pmatrix} 0 \\ \vec{x} \end{pmatrix} \bullet {}^N_B \dot{\mathbf{q}}_{E,k}, \end{aligned} \quad (5.15)$$

where  $\bullet$  indicates quaternion multiplication and  $\dot{\mathbf{q}}$  represents the conjugate quaternion to  $\mathbf{q}$ . The vector  ${}^B \vec{N}_{\text{IMU},k}$  now represents the direction as a substitute to the magnetometer-based north direction vector in the NED frame, as can be seen in figure 5.5.

With the complete substitute, it is possible to use a single GDA equation to compute the orientation estimate. The formula to calculate a complete measurement quaternion for the Kalman filter with either reference vector can be written as follows

$${}^N_B \mathbf{q}_{\nabla,t} = ({}^N_B \mathbf{q}_{\nabla,t-1} + {}^N_B \dot{\mathbf{q}}_{\omega} \Delta t) - \mu_k \frac{\nabla f_{a,m}}{\|\nabla f_{a,m}\|}, \quad (5.16)$$

where  $\frac{\nabla f_{a,m}}{\|\nabla f_{a,m}\|}$  is the GDA formulation to calculate a three dimensional quaternion from gravity and a heading vector, either  $\vec{N}_{\text{IMU}}$  or  $\vec{N}_m$ . Note that the equation includes a change of orientation from angular rate measurements expressed as quaternion ( ${}^N_B \dot{\mathbf{q}}_{\omega}$ ) to ensure fast convergence and omit overshoot from the GDA computation.

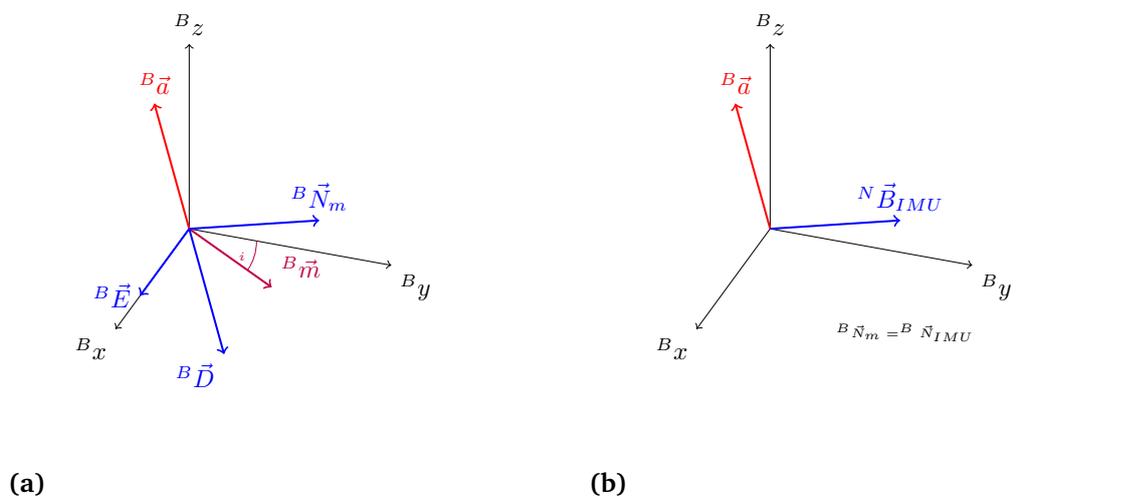
Since the vectors  ${}^B \vec{N}_m$  and  ${}^B \vec{N}_{\text{IMU}}$  lie in the same plane, it is possible to calculate a deviation angle ( $\epsilon$ ) that can be used to determine magnetic disturbance due to sudden changes in the direction of the north direction vector in contrast to the IMU heading vector. The deviation angle is calculated as follows

$$\epsilon = \cos^{-1} \left( {}^B \vec{N}_m \bullet {}^N \vec{B}_{\text{IMU}} \right), \quad (5.17)$$

where  $\bullet$  represents the dot product respectively.

The complete block diagram of the proposed filter is given in figure 5.6.

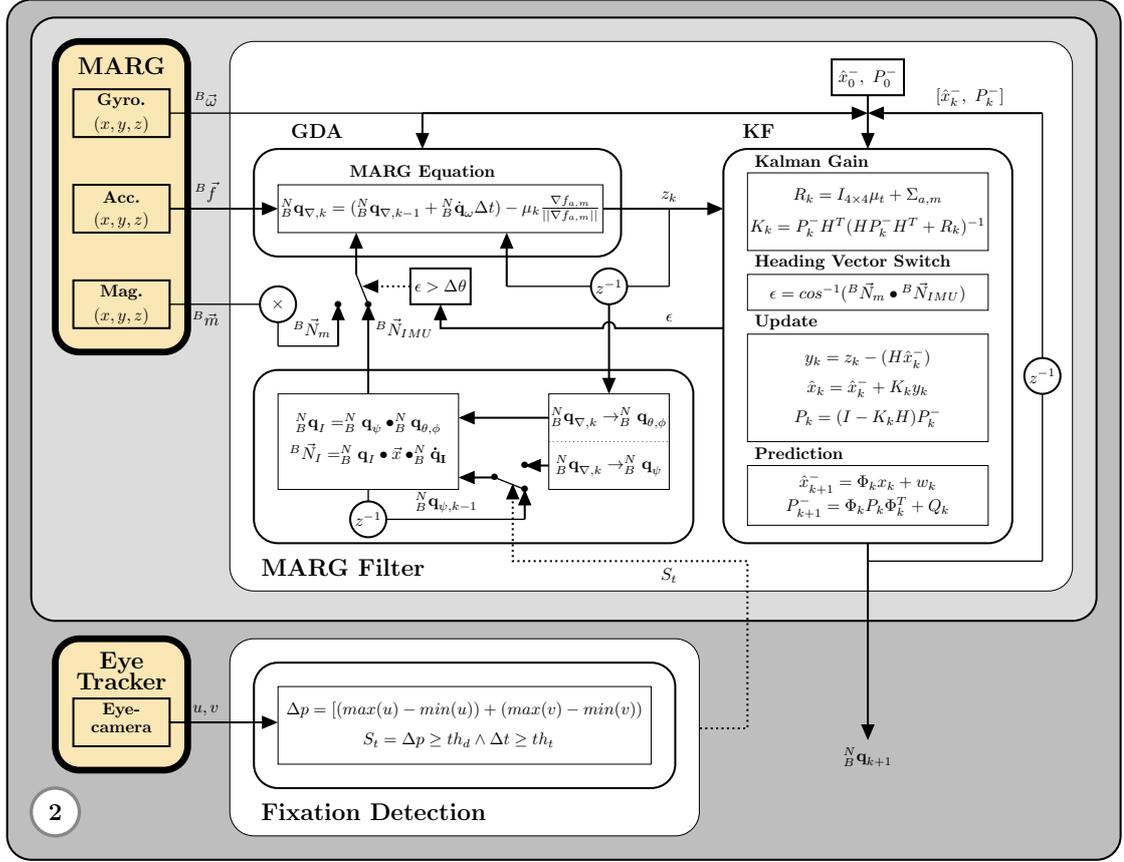
If magnetic disturbance is present, the deviation angle  $\epsilon$  will increase. If it exceeds a threshold  $\Delta\theta$ , the filter switches towards the virtual sensor vector-based quaternion calculation and vice versa if it vanishes. This procedure enables the calculation of a



**Figure 5.5:** Depiction of the north direction vector substitutes, (a) north direction vector, termed  ${}^B\vec{N}_m$  from accelerometer ( ${}^B\vec{a}$ ) and magnetometer ( ${}^B\vec{m}$ ) measurements in the case of undisturbed magnetic field measurement, and (b) IMU heading vector, termed  ${}^B\vec{N}_{IMU}$  calculated based on quaternion vector multiplication from the GDA filter without magnetometer data in the case of disturbed magnetic field measurements.

complete and continuous quaternion solution that involves current sensor measurements from the accelerometer and the extracted heading information from the previous quaternion. It is possible to use the same set of equations without any adaptation and switch from the magnetometer-based north direction vector to the IMU heading vector without divergence of the quaternion. While the zero-rotation trigger is enabled, the fusion process holds the recent calculated heading quaternion  ${}^N_B\mathbf{q}_{\psi,E}$ . This ensures that the GDA-based calculation of the new quaternion  ${}^N_B\mathbf{q}_k$  is less affected by possible drift in the heading direction due to uncorrected gyroscope bias but will however be corrected in the remaining axes through accelerometer updates and preserves a continuous solution and convergence. While no trigger is emitted, the fusion approach simply updates the measurement quaternion with every iteration based on either the magnetic north direction vector when no disturbance is present or the IMU heading vector from the current orientation.

Calculating the IMU heading vector instead of the prior implemented MARG/IMU switch has the rigorous advantage of being a lot less computationally expensive, since it is not necessary to calculate two full gradient quaternions but rather calculate only one gradient solution and only need four additional quaternion multiplications. This gives the possibility of calculating any other heading reference (rotated and transformed into the correct frame) to be used within the same set of equations and filter. The noise estimation for the measurement covariance matrix does not change drastically, since it is still majorly bound to the overall step size of the gradient that scales with respect to the



**Figure 5.6:** Block diagram of the eye tracking-supported MARG sensor data fusion approach. The fixation toggle enables a sample and hold vector substitute calculation which is recursively fed to the GDA-stage effectively reducing ARW and DC bias drift while the head is in a stationary motion condition.

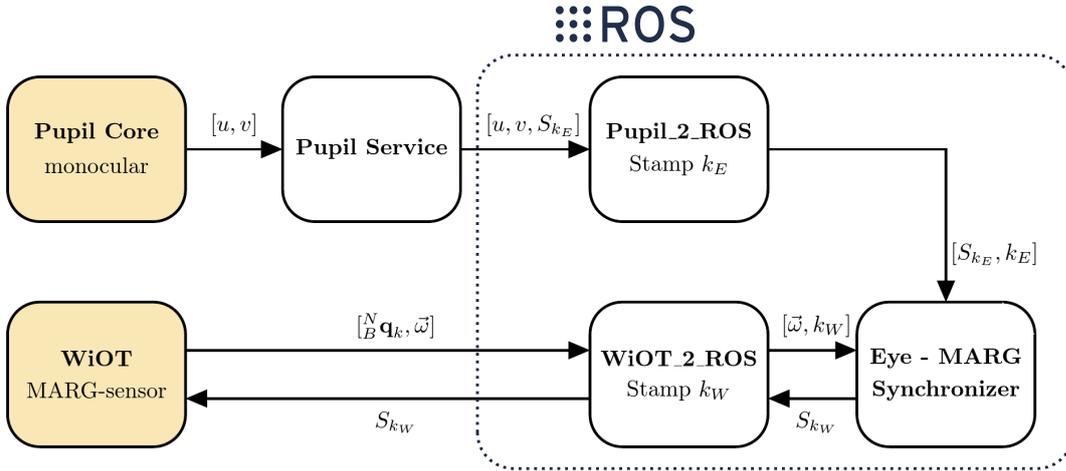
underlying motion. Under heavy dynamic motion the step size rises as well as the linear acceleration which will produce an erroneous reference quaternion. This quaternion should be less weighted in the orientation solution and will in any way be less weighted in the overall solution.

## 5.2 Setup and experimental results

The proposed data fusion process using visual fixations is implemented on the WiOT system. The MCU is running the FreeRTOS realtime operating system on both cores at 1 kHz scheduler tick rate [89]. The stand-alone implementation is designed to simultaneously calculate orientation data from two copies of the data fusion process at 300 Hz, while their only difference is the active eye tracking trigger update. The two data fusion filters run in real time on the MCU and publish the two sets of fused orientation data and the calibrated 9-axis sensor data at 100 Hz via Micro-USB over UART. The sensor is

## 5.2 Setup and experimental results

attached via a custom casing to the Pupil Core monocular eye tracker [54]. The tracker frame is secured via an eyewear strap on the user's head. The eye tracker is set to sample eye-camera frames at  $120\text{ Hz}$  at a resolution of  $400 \times 400$  pixels. It is connected to a host computer running the Pupil Labs open-source Pupil Service tool to acquire and preprocess the data as well as taking care of online fixation detection.



**Figure 5.7:** Data transfer and synchronization in ROS network between MARG sensor, eye tracker and ROS nodes.

The data is accessible in real-time through ZeroMQ. Two custom C++ ROS (Robot Operating System) nodes handle the synchronization and inter-device communication. Synchronization between the MARG and eye tracking data is achieved through comparison of their corresponding discrete timestamp ( $k_E$  for the eye tracker and  $k_W$  for the MARG sensor) upon arriving at the host system with a maximum lag of  $3\text{ [ms]}$  between the timestamps. While fixation is true, the trigger signal ( $S_{k_E}$ ) is broadcasted to the MARG sensor system indicating a zero-rotation. Furthermore, the trigger starts the gyroscope raw data ( $\vec{\omega}$ ) capture process on the host computer. When the visual fixation is released the trigger is set to false which stops the gyroscope capture process as well as the zero-orientation update cycle. The median gyroscope noise for stationary motion phases is sent to the MARG sensor, when sufficient amount of data has been captured. To reduce latency impact on the orientation calculation, a movement threshold based on this median gyroscope noise is implemented on the MARG sensor to ensure that the trigger will be set to false without latency drops or desynchronization. Figure 5.7 gives a simplified overview of the ROS nodes and software involved to synchronize the eye tracker and MARG sensor and transfer data in between them.

The accuracy of the proposed interface and the accompanying data fusion process is benchmarked against a Qualisys motion capture system (Qualisys Miquis Camera M3, Qualisys AB, Kvarnbergsgatan 2, 411 05 Göteborg, Sweden). The interface is worn by

a user alongside a lightweight 3D-printed rigid body passive IR-marker tree connected to the MARG sensor casing, see figure 5.8. The capture process of the Qualisys motion capture system broadcasts data at 120 Hz over a real-time client, allowing for timestamp based synchronization via the before mentioned ROS nodes. The threshold  $\Delta\theta$  is chosen based on the  $3\sigma$  standard deviation of the north direction vector under static conditions for a short period of time (10 s). The first calculated north direction vector of this series is the initial vector.



**Figure 5.8:** Experimental interface setup for visual fixation supported MARG sensor orientation estimation. The interface consists of the monocular Pupil Core head-worn eye tracker ①, the custom MARG sensor board (WiOT) ② that is rigidly connected to the frame and a passive IR-marker tree that is in line with the sensor ③.

This initial vector is used to calculate the standard deviation of this series of deviation angles  $\epsilon$  based on equation (5.17). For the ICM 20948 on the custom sensor board the threshold  $\Delta\theta$  results in  $3^\circ$ . After a warmup phase, the magnetometer data is turned off to simulate a magnetically disturbed environment and examine the eye tracking-supported zero-orientation change trigger update mechanism as a proof of concept. The proof of concept of the proposed orientation estimation update mechanism can be provided either by using real magnetometer data or by turning off the magnetometer data measurement completely. A difference is not evident. This is due to the switching from the north heading vector to the IMU heading vector when a magnetic disturbance is present. In such a case, magnetometer data is not used in the orientation estimation algorithm and is therefore not dependent on real magnetometer data input. Thus, disturbance is simulated by switching magnetometer data off to investigate the performance of the proposed data fusion during periods of non-usable magnetometer input. To compare

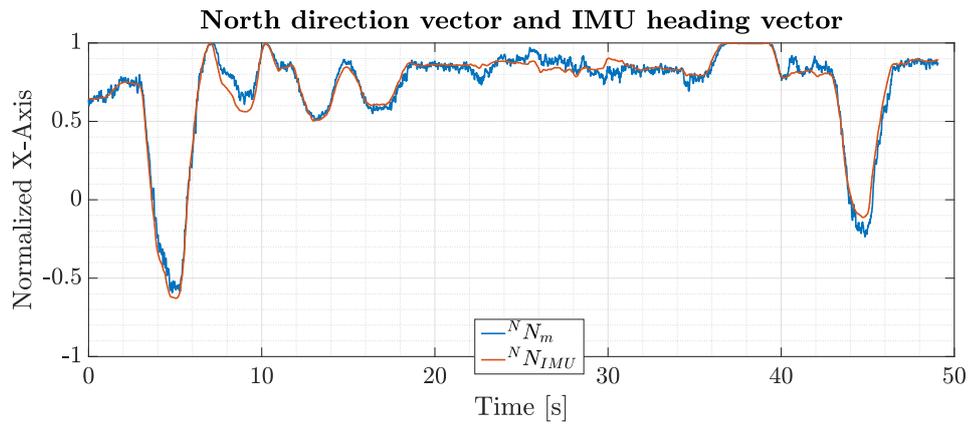
rotations, the coordinate system of the Qualisys data is transformed into the body coordinate frame of the MARG sensor by calculating an alignment matrix from six stationary positions through least square method described in reference [90]. The user is instructed to freely move his head and eyes with some static or no-rotation phases spanning between 2–5 s in duration. The total duration for one trial was limited to 15 minutes. A total of six trials was gathered for one individual user as a proof of concept for the proposed method. Visual fixation detection parameters were chosen based on experimental pretests that minimize latency drops when changing from stationary to dynamic head motion and were set to the following:  $th_d = 0.21^\circ$ , and  $th_t = 220$  ms.

Two pretests are conducted to demonstrate the interchangeability of the north direction vector substitute calculations described in section 5.1.2 and the filter's capability to detect interference based on the deviation angle  $\epsilon$ . The three-axis magnetometer is calibrated based on the process described in reference [77]. The magnetometer calibration is based on an ellipsoid to unit sphere mapping of a dense tri-axis magnetometer point cloud. The MARG sensor is rotated arbitrarily in all dimensions. The tri-axis magnetometer data is sampled during this period. After recording, the magnetometer data is calibrated through least-squares fitting of the ellipsoid data into a unit sphere and scaled to the surrounding field strength afterwards. During a 10-minute warmup phase, the filter uses magnetometer data to converge towards the direction of magnetic north and gravity respectively. To demonstrate the interchangeability of the vectors, the filter is switched off to use the IMU heading vector instead of the magnetometer-based north direction vector and move the sensor arbitrarily for a short period of time (50 s). Both vector measurements are recorded throughout the trial. The second pretest covers the validation of magnetic interference detection and the switching from north to IMU heading vector. The sensor is set up according to the previous mentioned calibration and warmup routines. Two sets of orientation are recorded: a) The proposed filter with magnetic interference detection and switching and b) the same filter without the switching mechanism. After 44 s an iron bar is brought close to the sensor (15 cm) to introduce magnetic interference.

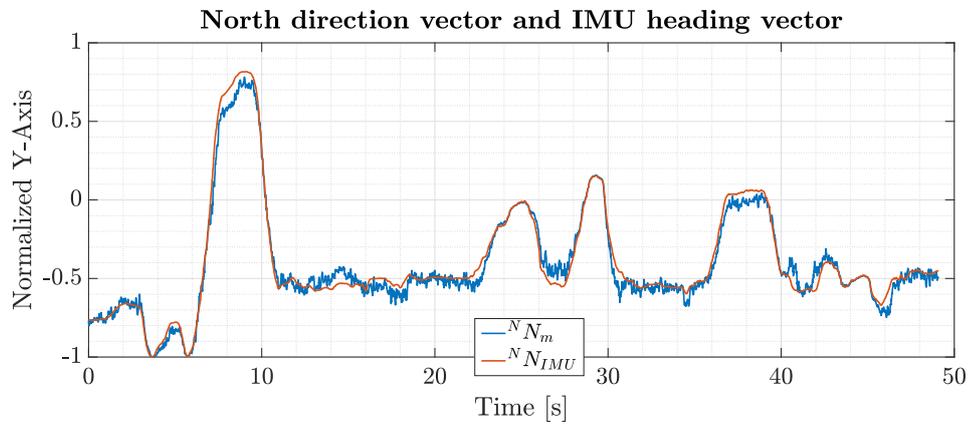
Subsection 5.2.1 gives an overview of the pretest to show the interchangeability of the heading vector substitutes calculation described in section 5.1.2 whereas subsection 5.2.2 examines the filter's capability to detect magnetic disturbance and switch towards IMU heading vector. Subsection 5.2.3 presents the experimental results from the full fusion approach using visual fixations for the zero-rotation update.

### 5.2.1 Interchangeable north direction vector substitutes

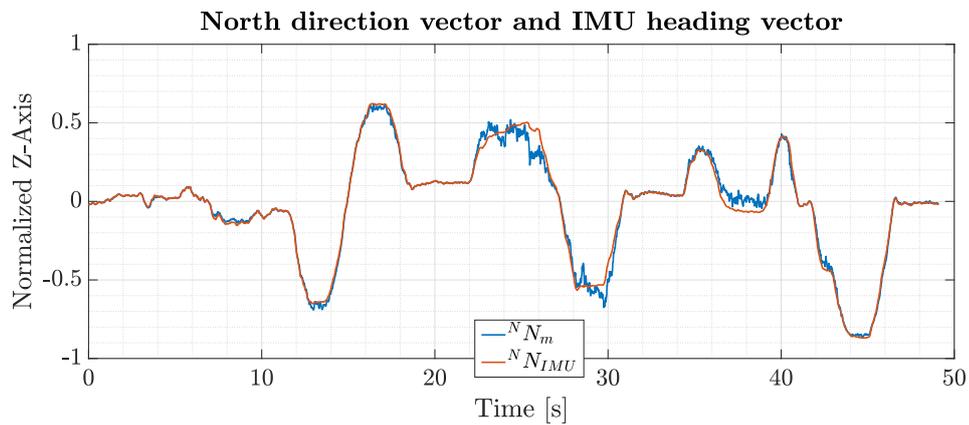
Figure 5.9 depicts normalized individual x-, y- and z-axis results for north direction vector  ${}^B\vec{N}_m$  from calibrated magnetometer data through equations (5.7) and the IMU heading vector  ${}^B\vec{N}_{IMU}$  based on the process given by equations (5.11)–(5.15).



(a)



(b)



(c)

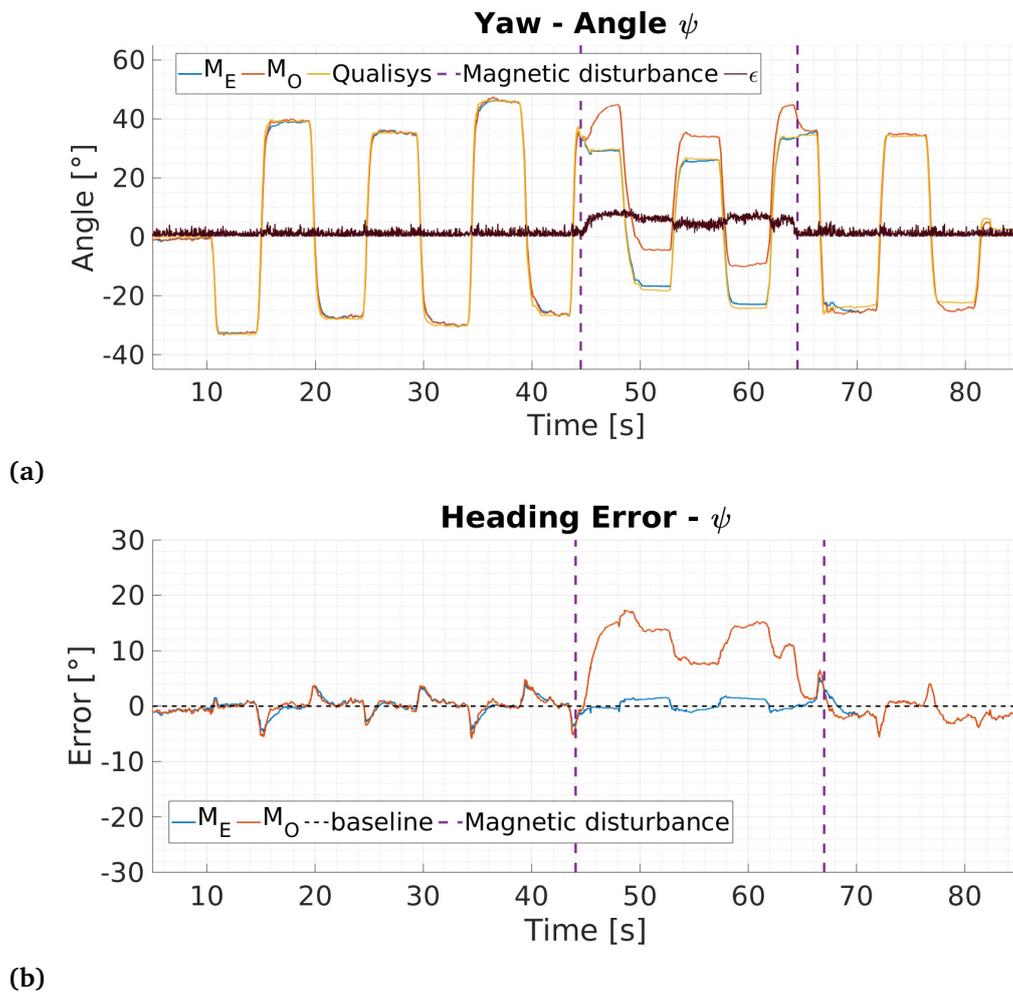
**Figure 5.9:** Comparison between (a) magnetometer-based normalized x-axis north direction vector and normalized x-axis IMU heading vector, (b) magnetometer-based normalized y-axis north direction vector and normalized y-axis IMU heading vector and (c) magnetometer-based normalized z-axis north direction vector and normalized z-axis IMU heading vector.

Both vectors show similar results during the whole trial with maximum deviations of  $\pm 0.1$  normalized units. The north direction vector of the magnetometer data has a larger variance of measurement values compared to the IMU heading vector. This originates from the different noise characteristics and computations of the vector components. The north direction vector is directly calculated from raw accelerometer and magnetometer data and will directly reflect raw sensor noise, whereas the IMU heading vector is based on smoothed quaternion fusion from gyroscope and accelerometer readings from the GDA filter. The noise spreading level, however, does stay at a reasonable level during the trial. It could however be smoothed by calculating the magnetometer-based heading vector through equation (5.8). This pretest shows the interchangeability of the different north direction vector substitutes which guarantees a continuous quaternion solution and convergence of the filter.

The IMU heading vector will drift apart from the north direction vector with respect to time. This is due to uncorrected gyroscope bias and ARW resulting in drift in the heading estimate of the quaternion used for determining the IMU heading vector. For short periods of time and under the same initial conditions, however, both vectors are almost identical. The length of the time period in which both vectors are mostly identical depends on the individual noise characteristics of the used gyroscope and computational errors from the discrete implementation. High-grade navigation gyroscopes will experience less drift compared to consumer-grade gyroscopes used in this work. The maximum time before gyroscope errors accumulate more than  $1^\circ$  drift in the heading estimate is 50 [s] for the custom MARG sensor board used in this work.

### 5.2.2 Magnetic disturbance detection

Figure 5.10 depicts yaw angle results as well as the corresponding yaw angle errors for the magnetic disturbance detection and switching from the north direction to IMU heading vector based on equation 5.17. The figure 5.10a presents three different yaw angle estimations over time: ground truth yaw data (Qualisys, yellow), yaw estimations from the proposed filter with deviation detection and heading vector substitutes ( $M_E$ , blue) as well as yaw estimations of a version of the filter without heading vector switching ( $M_O$ , orange). The figure also presents values for the magnetic deviation angle  $\epsilon$  (black) over time. Figure 5.10b presents the corresponding yaw angle errors referenced to the ground truth yaw data. Magnetic disturbance is introduced for a short period of time starting at 44 [s] and ending at 66 [s] by bringing an iron bar close to the sensor (15 [cm]). The proposed filter ( $M_E$ ) detects the disturbance when it is introduced because the deviation angle  $\epsilon$  exceeds the threshold  $\Delta\theta$ , see figure 5.10a. The filter switches towards IMU heading vector substitute and is not affected by the disturbance, resulting in a maximum error of  $2^\circ$  during this phase. In contrast, the filter without switching mechanism experiences large yaw angle errors and results in up to  $17^\circ$  total error (see figure 5.10b). This pretest demonstrates the filter's capability of magnetic disturbance detection based

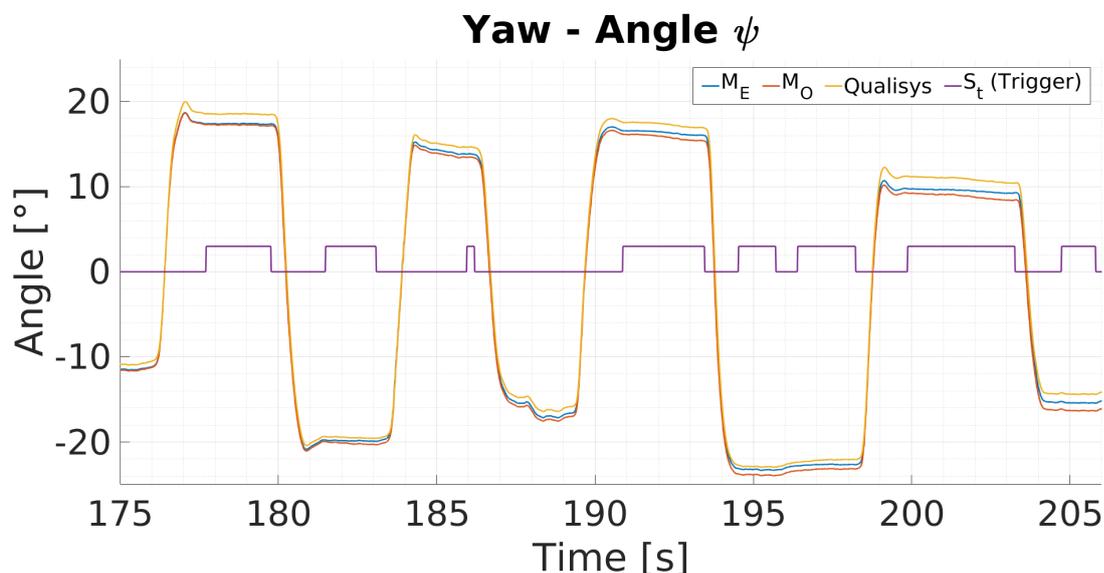


**Figure 5.10:** Influence of magnetic disturbance on yaw angle estimation: a) yaw angle comparisons between ground truth (Qualisys, yellow), the proposed filter with magnetic disturbance detection and heading vector substitute switching ( $M_E$ , blue), a filter version without heading vector switching ( $M_O$ , orange) as well as the magnetic deviation angle  $\epsilon$  (black). Figure b) depicts the corresponding heading error referenced to the Qualisys system. Magnetic interference is introduced for a short period of time (43 [s] to 66 [s]) by bringing an iron bar close to the MARG sensor. The proposed filter detects the interference and switches towards IMU heading vector usage resulting in less error.

on the deviation angle calculation between the north direction and the IMU heading vector. After the filter detects a disturbance, it uses the IMU heading vector for orientation estimation. In this mode the filter furthermore enables a visual zero-rotation update mechanism to reduce heading error accumulation over time.

### 5.2.3 MARG sensor data fusion approach using visual fixations

Figure 5.11 shows typical data for yaw angle measurements from a 30 [s] sequence of one 900 [s] trial. The yaw angles are presented in degrees over time in seconds. The figure depicts yaw angle estimation data from the ground truth motion capture system (Qualisys system, yellow), the proposed ( $M_E$ , blue) and standard version ( $M_O$ , orange) of the data fusion process. The visual fixation trigger state  $S_k$  is presented in purple. During visual fixation phases ( $S_k$ , purple) the proposed eye tracking-supported version  $M_E$  does drift less compared to the standard implementation  $M_O$ . In dynamic motion phases both filter versions do accumulate the same amount of drift.



**Figure 5.11:** Typical yaw angle measurements for a motion sequence from the Qualisys system (yellow), the proposed ( $M_E$ , blue) and standard version ( $M_O$ , orange) of the data fusion. During stationary motion, the trigger signal  $S_k$  (purple) is set to high and indicates zero-orientation change.

The performance of the proposed approach for the heading estimation is presented as total Euler angle error (degrees see figures 5.12) as well as mean error reduction rate ( $\epsilon_\psi$ , unit-less see Table 5.1). The total Euler angle errors are calculated as the difference between the ground truth of absolute orientation from the Qualisys system and the orientation estimation of the proposed ( $M_E$ ) and standard version ( $M_O$ ) of the data fusion process (see figure 5.11). The dashed black line indicates the reference for zero heading error. Figure 5.12 presents eight sets of Euler angle errors in degrees over time for the complete 900 [s] duration of two different trials. At the start of the trial the eye tracking-supported version of the filter ( $M_E$ , blue) as well as the standard GDA filter ( $M_O$ , orange) perform identical. During dynamic phases both filters accumulate the same amount of error due to uncorrected gyroscope bias and ARW noise. However, when stationary phases are indicated and the trigger signal  $S_k$  is enabled, the eye tracker-

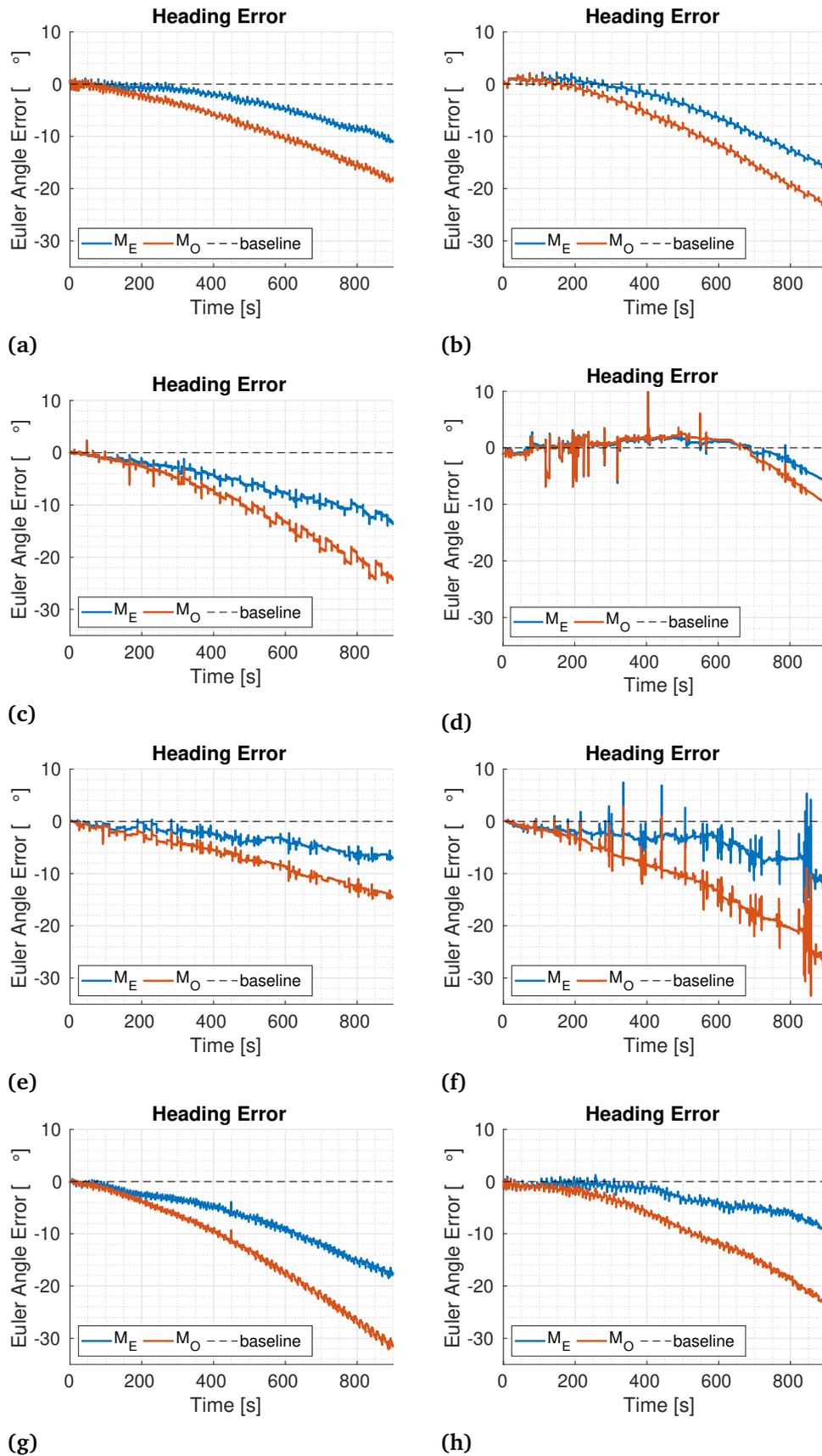


Figure 5.12: Error graphs for absolute heading error referenced to the Qualisys system from the GDA-based approach with ( $M_E$ , blue) and without eye tracking ( $M_O$ , orange) support for eight different trials.

### 5.3 Conclusion

supported GDA filter version accumulates less gyroscope drift in contrast to the standard implementation, see figure 5.12. This effect covers the entire duration of the trials. The orientation estimation errors rise significantly over time for both solutions. The GDA-based approach with eye tracking-based zero-orientation change update results in nearly 50 percent less absolute orientation, see any graph in figure 5.12. The different trials underline the potential drift reduction by using eye tracking stationary condition. The results may differ in both magnitude and their respective sign. This is due to different ARW and bias contributions for each new trial. Major but short-time errors are due to lagging from the synchronization step between the Qualisys and WiOT ROS node.

The mean error reduction rate  $\epsilon_\psi$  and its standard deviation is calculated based on the absolute error quotient between the proposed ( $M_E$ ) and standard version ( $M_O$ ) of the fusion process at 900 [s]. Table 5.1 presents total Euler angle errors from eight different trials for the proposed ( $M_E$ ) and standard version ( $M_O$ ) of the fusion after 900 [s] and the calculated error reduction rate. On average, the eye tracking-supported GDA filter approach accumulates near 50 percent less orientation error ( $0.46 \pm 0.07$ ) compared to the GDA filter without eye tracking data support, see table 5.1.

**Table 5.1:** Absolute error values and error reduction rate ( $\epsilon_\psi$ ) for the GDA based data fusion with ( $M_E$ ) and without eye tracking ( $M_O$ ) support after 900 [s].

Trial Nr.	$M_E$ [°]	$M_O$ [°]	$\epsilon_\psi = 1 - \frac{M_E}{M_O}$
1.	-18.84	-32.76	0.42
2.	-8.82	-22.55	0.59
3.	-13.89	-23.17	0.40
4.	-11.33	-19.09	0.40
5.	-12.63	-26.06	0.51
6.	-7.16	-13.52	0.45
7.	-6.268	-10.35	0.39
8.	-9.87	-24.37	0.59
average $\epsilon_\psi$			$0.46 \pm 0.07$

### 5.3 Conclusion

Utilizing eye tracking data to support sensor data fusion of MARG sensors shows improvements of the heading accuracy in magnetically disturbed environments or for IMU sensors that do not feature a heading reference in the order of 50 percent. Because of the physiological coupling between eye- and head-rotations, eye tracking can deliver an indicator signal for near-zero head orientation change. Furthermore, this trigger signal allows for individual and adaptive noise parametrization through gyroscope capturing

and could be used in the context of adaptive noise estimation with respect to head motion while a sufficient amount of data is captured. The proposed method can be used with any mobile eye tracking devices that either feature a built-in IMU or MARG sensor or are expanded by a custom or third-party sensor. The presented approach does not need a scene camera and is therefore mostly independent of surrounding environmental lighting conditions. In addition, the proposed use of interchangeable north direction vector substitutes enables switching between full MARG- and IMU-mode, without the need for an additional set of equations in a given filter. This furthermore solves the problem formulated in the initial filter framework that would arise from parallel computation of two quaternions. Upon switching both could diverge, but since the new formulation does compute one single measurement quaternion that converges to either reference vector from the same prior quaternion, the solution quaternion is continuous. This guarantees a continuous quaternion computation and convergence of the filter.

### 5.3.1 Limitations

The magnitude of error compensation does scale with respect to total fixation duration and amount of stationary motion phases. However, the solution does not reduce the effect during dynamic motion phases since it does currently not directly estimate and correct the DC bias term of the raw gyroscope signal. This is due to various other noise effects that are present in the raw gyroscope signal. Main noise terms among other that influence the in-run DC bias estimation, are ARW, oscillating head motion, output rate limitations and possible desynchronization between timestamps of both systems.

When estimating in-run DC bias, the presence of these noise terms can lead to a wrong estimation. Since the DC bias is subtracted from the raw gyroscope signal at every time step, it effects the complete measurement from that point forward and might result in a worse heading estimate. However, if a sufficient amount of sensor data has been gathered, a low-pass filtered DC bias estimation can be used to reduce the drift at a smaller scale since the data is only captured during near-stationary motion phases and therefore restrains heavy amplitude changes.

The proposed solution can be affected by very slow-motion acceleration triggering the visual fixation detection plug-in and falsely labelling a static phase. This effect, however, only appears if the resulting angular rate of the head motion is smaller than the angular rate constraint derived from the dispersion and time threshold of the fixation detection plug-in and stays below the median angular rate threshold that is sampled throughout the trial. In this work the angular rate constraint from the fixation detection plug-in that might lead to a wrong classification during fixation and simultaneous head motion is  $0.95 \frac{\circ}{s}$  for a  $220 [ms]$  measurement window. This angular rate results in the maximum dispersion of  $0.21^\circ$ . This would result in a fixation detection which would in turn trigger the zero-rotation update mechanism for one cycle. After this, the dispersion threshold is exceeded, setting the trigger to false which in turn resets the online fixation detection

sliding window.

Upon comparing this rotational velocity limit with a commercially available MARG sensor from section 4.4, this detection limitation is less than the slowest rotational speed that can be continuously resolved using the commercial device (slower than  $3^\circ/s$  results in a non-continuous stair like increase). The presented approach can increase the heading estimation robustness, but naturally scales with the amount of stationary motion phases. To ensure maximum heading estimation enhancements, one might introduce artificial fixations in the use case, e.g. when using the interface within a robot teleoperation workplace. An interaction designer might use fixations signals of the eye inside a GUI application on a monitor to trigger or start certain events. This will in turn increase the potential benefits of the proposed eye tracking support. This is nevertheless up to the designer.

The presented approach leans towards the goal of a robust and infrastructureless interface but can not quite deliver the needed safety measures and guarantee a robust heading estimate throughout all situations. To ensure a safe and reliable heading estimate further methods or references have to be researched. A possible solution is to employ computer vision methods (e.g. VO or vSLAM) on the scene camera images of the head-worn eye tracker and fuse these with the orientation estimations of the MARG sensor. A possible solution to this is given in the next chapter.

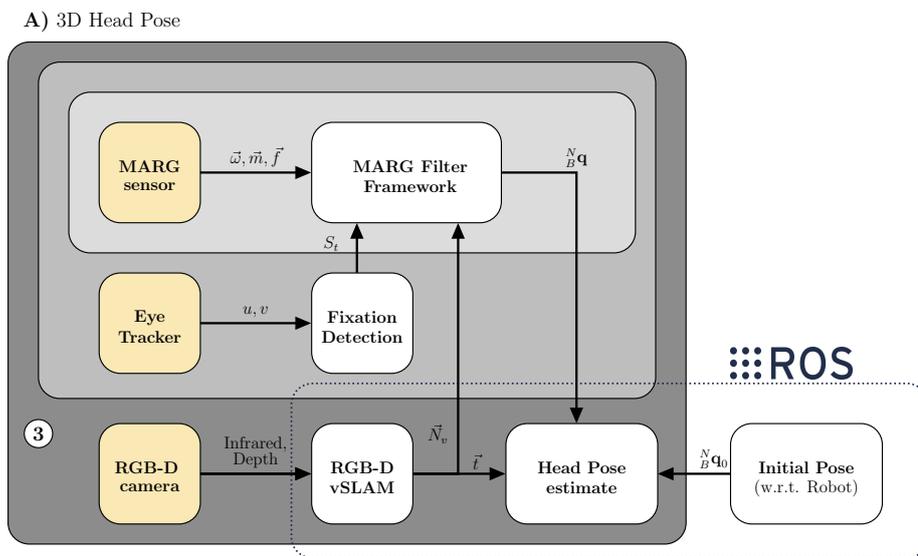


## 6 Visual SLAM-supported MARG sensor orientation estimation

The content and results of the following chapter were published in [5] and are extended by further details.

Using visual fixation detection of a head-worn eye tracker to support MARG sensor orientation estimation can reduce the accumulated gyroscope drift of commercially available MARG sensors. This, however, depends on the total amount of fixations detected, which cannot be foreseen at any point in time. Since head orientation estimation for robot teleoperation needs to be robust and precise to enable accurate and safe control mechanisms, additional measures need to be implemented.

An obvious approach is to use computer vision methods for pose estimation using the eye tracker's scene camera. Due to the technological advances in camera miniaturization and research efforts in computer vision over the last years, this chapter explores the usage of a loosely coupled vSLAM-supported MARG sensor orientation estimation approach, compare figure 6.1 for a simplified block diagram of the upcoming explanation



**Figure 6.1:** Simplified block diagram of the developed data fusion framework that is capable of fusing visual SLAM, eye tracking and MARG sensor data for robust pose estimation.

of the data fusion. Employing computer vision methods for a head-worn eye tracker contributes to a lightweight, infrastructureless solution for accurate head orientation estimation and furthermore provides the basis for additional control inputs. For the estimation of a robust head pose (orientation and position) with respect to the robot, it is possible to use head and eye gaze for 6-DoF robot control in Cartesian space to facilitate hands-free and multimodal human-robot collaboration.

The proposed data fusion addresses the environmental challenges which arise from the teleoperation task and is not reliably solvable using only a single sensing technology. For example, the orientation estimation of MARG sensors is based on the numerical integration of angular rate measured through the gyroscope. These low-cost, consumer-grade MEMS-based gyroscopes suffer from DC offsets, known as gyroscope bias. This bias leads to a drift in the integrated angles. This drift is typically compensated by using global references, i.e. direction of gravity and geomagnetic field of the earth measured by the accelerometer and magnetometer, in the data fusion process. However, the measurements are subject to external disturbances effecting the measured direction of the reference vectors and therefore leading to orientation estimation errors. This is especially the case for the measurement of the geomagnetic field vector used to correct heading estimation errors. Indoor scenarios and the presence of ferromagnetic and permanent magnetic materials (e.g. robotic systems) will lead to varying magnetic field vectors which degrade the effect from the geomagnetic vector measurement on the heading correction. Within this work visual odometry (VO) data is incorporated in the data fusion process of the MARG sensor to apply heading correction and improve absolute orientation estimation in indoor scenarios. On the other hand, the sole use of VO is not robust related to the proposed scenario. Using vision-based techniques only (e.g. optical flow) would also result in accumulations of errors since the visual scenery will be exposed to a lot of relative motion from the robotic system. Robust VO is based on the dominance of static feature points over moving objects and therefore degrades in the presence of moving objects, in this case the moving robot system.

## 6.1 Filter development

To address the aforementioned issues, this work utilizes a vSLAM approach to a) create an accurate map of the working area to relocalize within the map based on the detected and matched features in order to increase accuracy and robustness and b) fuse the measurements with MARG sensor data to reduce the impact of relative motion in front of the cameras visual scene on the orientation estimation and c) to be able to reset to a known orientation based on the discrete MARG sensor estimations. The head pose estimation block fuses visual and inertial sensor readings to form a robust pose estimation of a user's head without the need for external marker placement, i.e. fiducial markers. Due to the recent technology and research efforts in camera technologies, depth cam-

era sensors have become small, fast, reliable, and affordable when it comes to everyday use. Using a depth camera over a regular monocular 2D image sensor adds a completely new dimension and has major advantages when it comes to pose estimation in general. Therefore a stereo depth camera is used as the input measurement for a vSLAM approach that is combined with the orientation measurement of a MARG sensor's to generate reliable and robust orientation even under the condition of complete loss of visual information. The visual position estimation is used as true head position since MARG sensors are known for accumulating errors upon estimating translation from double integration of acceleration and might therefore lead to wrong position estimation.

### 6.1.1 Visual SLAM-based pose estimation

The vSLAM framework is part of block A, the head pose estimation pipeline in figure 3.1. In this work a state-of-the-art and open-source vSLAM approach, namely ORB-SLAM 2 [75], is used. This algorithm has proven to be very robust and precise regarding position and orientation estimation. The vSLAM approach uses ORB to detect and track features from an RGB-D input image.

In this work the RGB-D-based implementation of ORB-SLAM 2 is used but instead of supplying the RGB image, the infrared image of the first infrared camera sensor from the stereo infrared module of the Intel RealSense D435 camera is passed to the pipeline. This image is the basis for the depth image alignment. This avoids the need for an alignment step between RGB and depth image. Furthermore, the RGB camera is a rolling shutter whereas the infrared cameras are global shutter cameras and will therefore contain less motion blur. The infrared cameras also feature a wider Field Of View (FOV) compared to the RGB camera ( $H \times V \times D$  - Infrared:  $91.2^\circ \times 65.5^\circ \times 100.5^\circ$ , vs. RGB:  $64^\circ \times 41^\circ \times 72^\circ$  [76]). Using the infrared image decreases the data package throughput sent out by the camera. The D435 camera provides the possibility to toggle the laser emitter projector between two consecutive frames. Thus, one image frame is taken with and the next frame without the emitter projector. Images without emitter projector are used as the 2D-image input source for the ORB-SLAM framework, whereas the depth images are provided based on the depth estimations from the images with emitter projection. This is enabled by calculating the mean image brightness and selecting the image with lesser brightness as 2D-image source and the one with higher brightness as the depth source for an increased depth image density respectively. Additionally, this dense depth image is used as the input to the gaze mapper, compare figure 3.1.

Figure 6.2 shows the image pipeline outputs with and without emitter projector for infrared and depth streams. Using the above-described image pipeline decreases the necessary data package size that needs to be handled by the host computer to the infrared and aligned depth stream only. Furthermore, this procedure ensures a wider FOV image for the vSLAM algorithm and gaze mapping instead of using the RGB image. The vSLAM framework will locate and track ORB (Oriented Fast and Rotated BRIEF) features in the

current infrared and associated depth image and inserts keyframes into a new map. Based on epipolar geometry and fundamental matrix the camera pose (orientation and position) is estimated using a constant velocity model between consecutive frames and optimized through bundle adjustment. ORB-SLAM is capable of loop closure during map creation and furthermore relocalizes the camera when revisiting known locations inside the generated map [75]. The ORB-SLAM framework features a localization only mode to reduce computational costs that can be enabled if a sufficient map has been captured. This algorithm is capable of generating reliable position and orientation data while visual frames are available. If a sufficient large map has been created ( $> 50$  keyframes), the localization only mode is enabled to reduce computational costs and reduce pose estimation errors from relative motion in the visual scenery. To further enhance robustness, the visual odometry constant velocity motion model tracking is disabled in the localization mode such that the algorithm relies on pose estimation from matched feature points in the map only. On the one hand, this procedure ensures that the visual pose estimation is less error-prone to relative motion in the scene. On the other hand, the overall tracking capability of ORB-SLAM is reduced which will result in localization failures during strong dynamic motion. If ORB-SLAM fails to localize in the scene, the mapping mode is enabled again. The vSLAM framework passes the visual orientation data to the MARG framework for data fusion purposes described in section 6.1.2. The vSLAM framework directly provides the head position data to the complete head pose estimate, see figure 3.1. There is no need to fuse the position data from the vSLAM framework, as the position estimation from the MARG sensor is not reliable due to drift. The immanent DC bias of the accelerometer leads to a second order drift phenomenon based on the double integration of the accelerometer's raw sensor data. The orientation estimation from the vSLAM approach, however, is passed to the MARG-filter framework to calculate a reliable orientation even when visual feedback is lost or compromised by relative motion, e.g. the robotic system moving through the scenery, to bridge the gap and reinitialize the vSLAM algorithm.



**Figure 6.2:** Image sequence as seen by the filtered camera outputs. Left, image without emitter projector, middle, image with emitter projector pattern (white spots) and right, depth image (colored).

### 6.1.2 Visual-inertial orientation fusion

The second algorithm to robust pose estimation utilizes MARG sensor measurements fused with visual heading information from the vSLAM approach to calculate an orientation estimation. The 3D head pose block (A) of figure 6.1 depicts the data transfer of visual heading information  ${}^B\vec{N}_v$  from the RGB-D odometry framework to the MARG-filter framework, enabling the calculation of a full quaternion  ${}^N_B\mathbf{q}$  representing the user's head orientation estimation. Even if visual information is lost for a longer period of time (up to 25 s) or if the visual information is degraded because of high dynamic relative motion in the scene, e.g. the moving robotic system, data fusion allows for a robust orientation estimation.

The MARG filter framework transfers the iteratively updated quaternion, calculated at the on-board MCU of the MARG sensor, to the RGB-D filter framework. The quaternion is split into two separate quaternions: A roll-and-pitch quaternion and a yaw quaternion. The roll-and-pitch quaternion is directly used as true input for orientation information, whereas the yaw quaternion is corrected within the RGB-D filter framework using the vSLAM based heading vector. This is because the yaw quaternion is subject to drift originating from the dynamic gyroscope offset. The gyroscope offset drift in heading direction is corrected by applying a set of equations to calculate the visual heading vector within the RGB-D SLAM node. This process is based on the set of equations to calculate an IMU heading vector and apply it to a GDA-based filter while magnetic disturbance is present, see [4] or section 5.1.2. The IMU heading vector substitutes the magnetometer vector and therefore reduces the needed sets of equations and guarantees convergence as well as a continuous quaternion solution to the minimization problem. The filter uses the updated form of Madgwick's GDA approach, given in equations (4.14)–(4.17), which decouple the heading vector update from the pitch and roll update calculation and therefore enhances robustness when the heading vector is disturbed. The process to calculate the visual heading vector substitute is as follows.

The quaternion  ${}^N_C\mathbf{q}_{V,k}$  from the vSLAM algorithm is transformed into the MARG sensor body orientation through two-sided quaternion multiplication

$${}^N_B\mathbf{q}_{V,k} = {}^N_B\mathbf{q}_0 \bullet \left( {}^C_B\mathbf{q}_{rig} \bullet {}^N_C\mathbf{q}_{V,k} \bullet {}^C_B\dot{\mathbf{q}}_{rig} \right), \quad (6.1)$$

where  ${}^C_B\mathbf{q}_{rig}$  is the rigid transformation from the camera's coordinate frame to the MARG sensor's body frame,  $\dot{\mathbf{q}}$  represents the conjugate quaternion respectively and  ${}^N_B\mathbf{q}_0$  is the initial quaternion that aligns the visual orientation estimation in the navigation frame with the MARG sensor orientation.

The heading part of the transformed visual quaternion is extracted using equations (5.11)–(5.12). Substituting this into the equation yields

$$\begin{aligned} \mathbf{q} &= \left( ((q_{V,k,1}^2 + q_{V,k,2}^2 - q_{V,k,3}^2 - q_{V,k,4}^2)) \quad 0 \quad 0 \quad (2 \cdot (q_{V,k,2} \cdot q_{V,k,3} + q_{V,k,1} \cdot q_{V,k,4})) \right)^T, \\ \mathbf{q}_\psi &= \frac{\mathbf{q}}{\|\mathbf{q}\|} \\ \mathbf{q} &= \mathbf{q}_\psi + \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}^T, \\ \mathbf{q}_{\psi,v} &= \frac{\mathbf{q}}{\|\mathbf{q}\|}. \end{aligned} \quad (6.2)$$

Likewise to the process in section 5.1.2 a roll-and-pitch quaternion is calculated. This quaternion is based on the iteratively updated orientation estimation from the MARG sensor by conjugate quaternion multiplication of the heading quaternion from the MARG sensor and the current output quaternion, to get rid of the heading rotation

$$\mathbf{q}_{\phi,\theta,k} = \mathbf{q}_{\psi,k} \bullet \mathbf{q}_k. \quad (6.3)$$

A new quaternion is formed that represents the complete visual heading quaternion by quaternion multiplication from the visual heading and the roll-and-pitch quaternion

$$\mathbf{q}_{vh,k} = \mathbf{q}_{\psi,v} \bullet \mathbf{q}_{\phi,\theta,k}. \quad (6.4)$$

This quaternion is now used to directly transform an x-axis unit vector into the visual heading vector by quaternion multiplication

$$\begin{aligned} \vec{x} &= \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T \\ \begin{pmatrix} 0 \\ \vec{N}_{v,k} \end{pmatrix} &= \mathbf{q}_{vh,k} \bullet \begin{pmatrix} 0 \\ \vec{x} \end{pmatrix} \bullet \mathbf{q}_{vh,k}^{-1}. \end{aligned} \quad (6.5)$$

The visual heading vector  $\vec{N}_{v,k}$  is used as a complete substitute to the magnetometer north heading vector inside the GDA stage forming a complete and continuous quaternion solution. The quaternion from the GDA is now applied as measurement inside the update step of the linear Kalman filter to correct for orientation accumulation errors from gyroscope bias, see figure 6.3.

The mechanism of calculating a substitute for the GDA heading vector is not limited to the visual heading vector substitute (compare section 5.1.2). In the case of degraded data from the RGB-D odometry framework, e.g. visual occlusion, the procedure enables the use of IMU or magnetometer data for the calculation of the heading vector. Switching in between the three heading vector modes based on visual, magnetic, and inertial data respectively allows robust heading estimation based on the current availability and reliability of the different sensor measurements. The following text passage briefly explains the process for inertial and magnetic heading vector calculation, which

has already been presented in 5.1.2.

The method for calculating the heading vector  ${}^B\vec{N}_{\text{IMU}}$  given by IMU data is similar to that of the heading vector  ${}^B\vec{N}_v$  given by visual data. This is achieved by substituting  ${}^N_B\mathbf{q}_V$  in equation (6.2) with the Kalman filter output quaternion  ${}^N_B\mathbf{q}_k$  and calculating the IMU heading vector through equations (6.3)–(6.5). The process isolates the heading component for the transformation quaternion in equation 6.4 which allows to sample and hold the current heading orientation if heading rotation is not exceeding a certain motion condition, e.g. angular rate slower than  $0.01^\circ s^{-1}$ .

The magnetic heading is calculated based on the cross product between the measured gravity and magnetic field vectors from the MARG sensor either through equation (5.7) or (5.8).

Regardless of whether the heading vector is calculated based on visual, IMU or magnetic data it represents redundant information perpendicular to the plane defined by the pitch-and-roll component. However, in the case of disturbance of any data sources, the other sensors are used to calculate the heading vector. The result is a robust complete orientation estimation under various disturbances.

The filter switches between the heading sources based on vector scalar product comparisons with the current output of the Kalman filter heading estimate

$$\epsilon_h = \cos^{-1} \left( {}^B\vec{N}_k \cdot {}^B\vec{N}_h \right), \quad (6.6)$$

where  $\cdot$  represents the scalar product and  ${}^B\vec{N}_h$  is to be substituted with either visual, magnetic or IMU heading vectors. Based on the relative deviation of  $\epsilon_h$  the filter switches towards the appropriate heading vector input

$${}^B\vec{N}_k = \begin{cases} \vec{N}_v & \text{if } (\epsilon_v < \epsilon_m \wedge \epsilon_v < th_i) \\ \vec{N}_m & \text{if } (\epsilon_m < \epsilon_v \wedge \epsilon_m < th_i) \\ \vec{N}_{\text{IMU}} & \text{otherwise} \end{cases} \quad (6.7)$$

The filter selects the most reliable heading source from relative deviations  $\epsilon$  and availability of the heading sources in the current conditions. The fusion process presented in this work ensures a fast, robust, and continuous quaternion solution to be found under various disturbances of data sources. The filter selects the visual heading source under static and slow dynamic motion conditions since it delivers accurate heading information and is capable of correcting drift accumulation. During fast dynamic motion IMU measurements are selected as heading vector information. This is because of the vSLAM motion estimation artefacts caused by latency issues of the vSLAM pipeline due to the low sampling frequency of the camera measurements  $30 \text{ Hz}$ . The filter switches towards either the magnetic or IMU heading vector, if the visual heading is not available. If the visual heading vector is lost due to the vSLAM framework's inability to relocalize in the map within five seconds, the vSLAM mapping process is reset. During this time, the filter

relies on MARG sensor data only. Once the vSLAM algorithm is restarted, the current MARG sensor orientation is sampled and used to transform and align the orientation estimation into the common navigation frame of the MARG sensor

$${}^N_B \mathbf{q}_0 = {}^N_B \mathbf{q}_{k+1}. \quad (6.8)$$

The visual orientation estimation is transformed into the MARG coordinate frame based on equation (6.1).

The presented data fusion process is implemented on the custom MARG sensor's MCU running at 300 Hz ensuring low latency between data fusion updates and MARG sensor measurements, while only incorporating visual feedback into the filter if it meets the aforementioned motion conditions.

### 6.1.3 Visual position estimation

The visual position estimation of the vSLAM algorithm is transformed into the MARG sensor navigation frame based on the fused orientation estimation from the visual inertial orientation estimation, see section 6.1.2.

The translation vector is transformed into the MARG navigation frame based on the following process

$$\begin{aligned} \begin{pmatrix} 0 \\ {}^N \vec{t}_{r,k} \end{pmatrix} &= {}^N_B \mathbf{q}_{rig} \bullet \begin{pmatrix} 0 \\ {}^B \vec{t}_k \end{pmatrix} \bullet {}^N_B \dot{\mathbf{q}}_{rig}, \\ \begin{pmatrix} 0 \\ {}^N \vec{t}_{h,k} \end{pmatrix} &= {}^N_B \mathbf{q}_0 \bullet \begin{pmatrix} 0 \\ {}^N \vec{t}_{r,k} \end{pmatrix} \bullet {}^N_B \dot{\mathbf{q}}_0 + \begin{pmatrix} 0 \\ {}^N \vec{t}_0 \end{pmatrix}, \end{aligned} \quad (6.9)$$

where  ${}^N \vec{t}_0$  is the last known position estimate that is sampled upon reset of the mapping process

$${}^N \vec{t}_0 = {}^N \vec{t}_{k+1}. \quad (6.10)$$

Equations (6.1) and (6.9) describe the full pose transformation from camera to MARG sensor coordinate frame which is denoted as  ${}^N_B T$  for readability (see figure 6.3).

The fusion approach allows for robust orientation estimation, even if visual feedback is lost or magnetic disturbance is present, and therefore enables robust head pose estimation which is key for mobile, accurate and safe robot control. The approach builds upon on the interchangeable heading vector substitution mechanism presented in chapter 5. It is therefore possible to use the developed methods for estimation accuracy enhancements alongside each other and contribute to a stable orientation estimation result.

The estimated head pose in the MARG frame is transformed into the robots coordinate system to allow for direct orientation and position estimation in the applications Cartesian space.

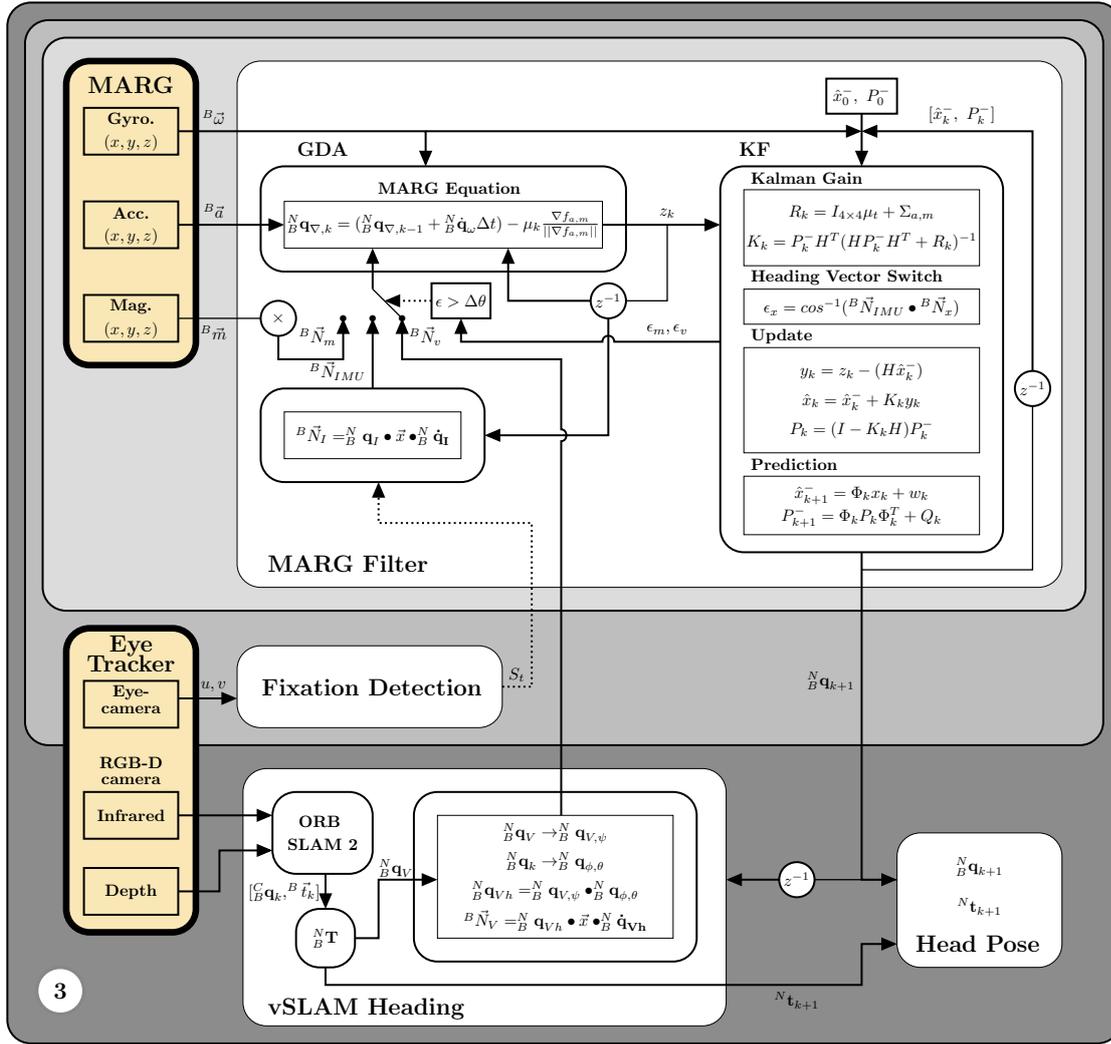


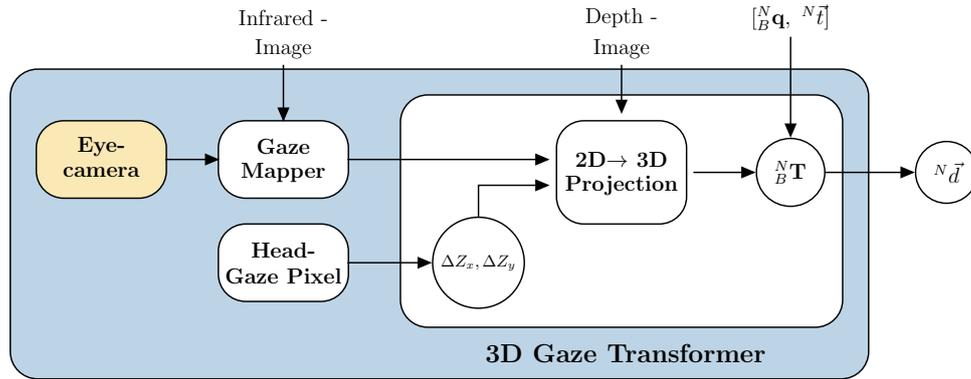
Figure 6.3: Block diagram of the complete data fusion process to incorporate either vSLAM, visual fixation or magnetic heading reference vectors into the core filter framework.

## 6.2 Beyond head motion – 3D gaze-based robot control

Since the interface enables an estimation of the head pose in three-dimensional space with respect to the robot coordinate system, it is possible to estimate a three-dimensional gaze point as a control input for a robotic system.

The proposed interface is designed to generate two different gaze signals: eye gaze and head gaze, respectively. First eye gaze mapping is described followed by gaze transformation, see figure 6.4 upper part. Secondly head gaze mapping and the respective real-world transformation is described, see figure 6.4 lower part.

Obtaining accurate eye gaze data strongly depends on the eye-to-scene camera calibration. Three-dimensional eye gaze estimations from binocular gaze ray intersection are



**Figure 6.4:** Block diagram of the 3D gaze point estimation pipeline. The eye camera’s measurements are mapped onto the infrared stream to generate a pixel pair whereas the head pose pixel pair is a fixed value. Both pixel pairs are passed to the gaze transformer to reproject the 2D pixel to 3D local camera coordinates. This local vector is lastly transformed into the world coordinate system forming  ${}^N\vec{d}$ .

heavily dependent on the eye model and calibration accuracy [59]. Instead of using a 3D-gaze vergence model between both eye tracker cameras, a standard 2D-calibration based on polynomial mapping is used to calibrate binocular pupil positions onto the scene camera’s image. The gaze mapper tracks a fiducial marker at five different locations (e.g., on a computer screen or presented by hand) and samples gaze pixel coordinates from the eye-cameras alongside world pixel coordinates of the fiducial marker. The parameters of the second-degree polynomial are calculated from standard five-point target calibration using singular value decomposition for binocular gaze data [51]. The gaze mapper consists of two custom ROS nodes that synchronize the pupil detection results with the Intel RealSense D435 infrared image stream and furthermore enable the AprilTag-based eye gaze calibration routine. This procedure ensures, that the RealSense camera port is not blocked by a single application and can be accessed by all nodes inside the ROS network, i.e. ORB-SLAM node, Pupil Service and infrared image synchronization as well as AprilTag detection. Pupil Labs officially cancelled support for the RealSense Depth Cameras due to the ongoing changes of the "librealsense" libraries and dependencies which make integration into the Pupil Capture framework rather difficult (OS- and Kernel-dependent). It is, however, possible to include a custom plug-in to use the RealSense depth or infrared streams in Pupil Capture.

This work, however, does not rely on this integration or custom plug-in. During this thesis, I have implemented a stand-alone ROS node that uses Intel’s official RealSense ROS-package, a Pupil Labs to ROS node package and a graphical calibration node based on AprilTag to achieve the gaze mapping with all the latest software from Pupil Labs and Intel RealSense (version independent). This also enables a direct transfer of the different image streams to more instances of nodes subscribing to the image transport

topic instead of blocking the physical port for other applications. Instead of using Pupil Capture, this work used Pupil Service which only streams pupil detection results without any scene camera stream (or calibration). The camera’s frame and the pupil detections are synchronized using the ROS `message_filter` package and the calibration of the eye gaze is done via a custom built GUI application that uses AprilTag markers for as a gaze calibration routine.

In this work a lightweight binocular eye tracker with an USB Type C mount from Pupil Labs and Pupil Service [54] is used. The Pupil Service client provides the pupil detection pipeline which is then used inside the gaze mapper. The gaze mapper uses the filtered 2D infrared image stream (no emitter, see figure 6.2) as the calibration target image. The calibrated 2D gaze pixel coordinates on the 2D infrared image are used to get the gaze vectors magnitude from the aligned 3D stereo depth image. This single real world depth value is transformed into a 3D vector  ${}^B\vec{d}$  in the camera coordinate frame by using point cloud reconstruction from the 2D pixel coordinates alongside the real-world depth value into a 3D point using the camera’s intrinsic parameters.

Using the pinhole camera model without lens distortion (see figure 6.5a, a 3D point is projected into the image plane using a perspective transformation

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (6.11)$$

Leaving out the perspective transformation, assuming the camera coordinate is the origin, the formula can be rewritten to the following:

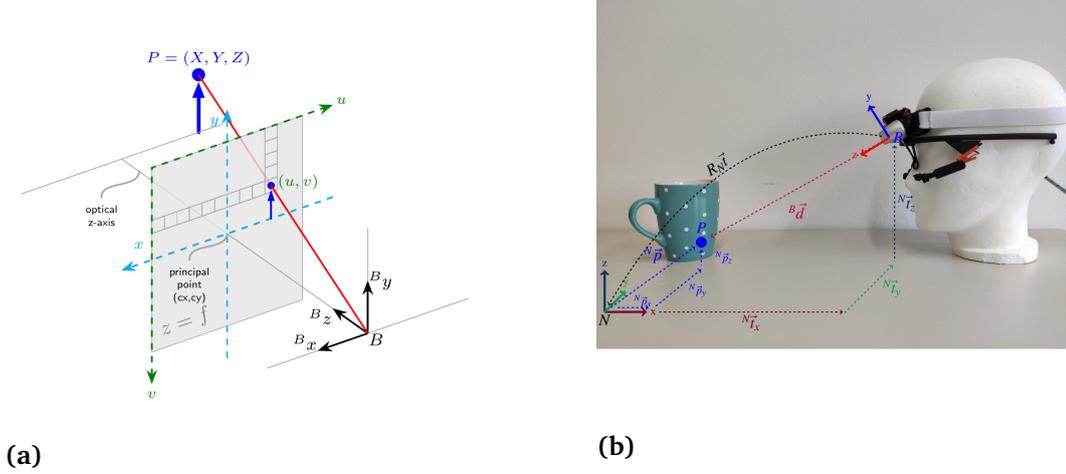
$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (6.12)$$

Since we get the pixel coordinates  $u, v$  in the infrared stream from the mapped gaze point we can directly select the real world depth value  $Z$  from the depth image stream which is aligned to the infrared image stream. Having  $u, v$  and  $Z$ , equation (6.12) can be rearranged and reduced to get the  $X$  and  $Y$  coordinates

$$\begin{aligned} X &= \frac{(u-c_x)*Z}{f_x}, \\ Y &= \frac{(v-c_y)*Z}{f_y}, \end{aligned} \quad (6.13)$$

$${}^B\vec{d} = \begin{pmatrix} X & Y & Z \end{pmatrix}^T.$$

Head pose estimate and gaze mapper outputs are the input variables for the 3D gaze transformer, see figure 6.4.



**Figure 6.5:** 3D gaze point reconstruction: (a) pinhole camera model, adapted from [91] and (b) illustration of gaze point ( $P$ ) depth vector ( ${}^B\vec{d}$ ) coordinate transformation from body ( $B$ ) to world coordinate frame ( $N$ ), where  $N$  is either coincident with the robot's origin or the transformation from  $N$  to the robot's origin is known and incorporated into the inverse kinematic chain. The vector  ${}^N\vec{p}$  is the input target point for the inverse kinematic calculation of the robotic system.

The 3D vector  ${}^B\vec{d}$  is transformed from the local camera coordinate system into the world coordinate frame by using the perspective transformation which is the estimated head pose ( ${}^N\mathbf{q}_k, {}^N\vec{t}_k$ ) in the robot's coordinate frame from the visual-inertial pose estimation, see figure 6.5b. The full head pose transformation from MARG sensor to robot coordinate frame is given through equations (6.1) and (6.9), substituting the static quaternion ( ${}^N\mathbf{q}_{rig}$ ) with the one that transforms from the MARG sensor to the robot's coordinate frame and set the initial alignment pose ( ${}^N\mathbf{q}_0, {}^N\vec{t}_0$ ) in the robot's coordinate frame. This transformation can be given either by providing an absolute position estimate in the robots coordinate system, e.g., using fiducial marker detection, or by using a three-point initial setup routine that defines the robot's coordinate system. To perform initial pose estimation, the user needs to focus the laser at three dots to define the x- and z-axis of the coordinate frame center that are known in the robot's coordinate system – e.g., focusing the base of the robot to align coordinate frames. The y-axis is calculated from the cross product between the defined x- and z-axis.

Substituting the rotation matrix from the perspective transformation by two-sided quaternion multiplication results in the following formula for 3D world gaze estimation

$$\begin{pmatrix} 0 \\ {}^N\vec{d} \end{pmatrix} = {}^N\mathbf{q}_k \bullet \begin{pmatrix} 0 \\ {}^B\vec{d} \end{pmatrix} \bullet {}^N\mathbf{q}_k + \begin{pmatrix} 0 \\ {}^N\vec{t} \end{pmatrix}, \quad (6.14)$$

$${}^N\vec{d} = (x \ y \ z)^T.$$

Using the above-mentioned setup allows for accurate 3D eye gaze estimation in a working area, that is restricted by the RGB-D camera's depth resolution. The gaze mapping method will only correspond to the pixel value chosen.

The interface can also be used for head gaze-based 3D point estimation. In this mode, a small eye-safe laser mounted above the camera is used for direct user feedback. The laser indicates the head gaze vector endpoint that is used for vector magnitude estimation and world point transformation. The laser pointer's pixel position in the camera frame is calculated based on equations (6.13) and (6.14). Since the laser's coordinate system and camera coordinate system are not perfectly aligned, the physical displacements and rotation from the laser pointer with respect to the camera center and projection of the depth value on the surface need to be incorporated. The depth value  $Z$  from vector  ${}^B\vec{d}$  is calculated based on the known orientation of the camera with respect to the world coordinate frame and is projected using trigonometric relations of the predicted real-world coordinates of the laser dot with respect to the camera frame ( $B$ ). This results in the following set of equations to calculate the projected depth  $Z_l$  from the center pixel ( $px, py$ ) and the x and y-axis displacement of the laser pointer,

$$Z_l = Z(px, py) + \Delta Z_x + \Delta Z_y \quad (6.15)$$

where  $\Delta Z_x$  and  $\Delta Z_y$  are calculated as

$$\begin{aligned} \Delta Z_x &= \frac{X_l}{\tan(\pi-\theta)} \\ \Delta Z_y &= \frac{Y_l}{\tan(\pi-\psi)}, \end{aligned} \quad (6.16)$$

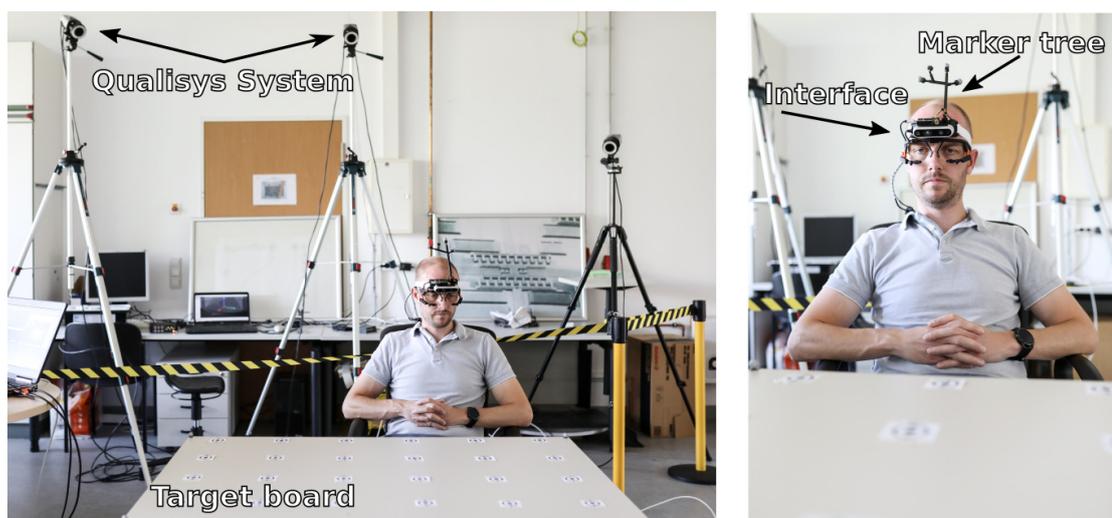
$X_l$  and  $Y_l$  are the physical displacements between the laser pointer's center and the camera's center, while  $\theta, \psi$  are the pitch and yaw Euler angles acquired from the cameras' orientation in the world frame ( $N$ ), respectively. The laser pointer's pixel position is calculated based on equation (6.13) where the input vector  ${}^B\vec{d}_l$  is

$${}^B\vec{d}_l = \begin{pmatrix} X_l & Y_l & Z_l \end{pmatrix}^T. \quad (6.17)$$

### 6.3 Setup and experimental results

The experimental setup is designed to evaluate a) the long-term heading drift reduction through the proposed visual and inertial data fusion, in contrast to inertial or visual data only orientation estimation, and b) the short-term orientation estimation stability if visual data is not available.

The accuracy of the pose estimation with the proposed interface is benchmarked against an infrared-based marker system (Qualisys Miquis Camera M3, Qualisys AB, Kvarnbergsgatan 2, 41105 Göteborg, Sweden). Therefore, the user is wearing the interface along-



**Figure 6.6:** Experimental setup. The user is wearing the interface and is sitting at a table in front of a Qualisys motion capture system. The user points at targets by either head gaze or eye gaze. The interface is equipped with a rigid marker tree that is attached on top of the custom camera case to capture ground truth data.

side a rigid marker-tree on top of the 3D printed custom case, see figure 6.6. The user is sitting in front of a table with a surface area of  $0.8\text{ m}$  by  $1.2\text{ m}$ . The table's surface is covered with a target marker grid representing the head and eye gaze targets. A total of 24 targets is placed on the surface, spaced evenly in a  $0.2\text{ m}$  by  $0.2\text{ m}$  grid, see figure 6.6. The target positions are known with respect to the world coordinate system from an initial measurement using the Qualisys reference system. The current pose of the rigid marker tree is used as the initial pose of the interface and is passed to the head pose estimation pipeline to align the interface pose with the Qualisys pose. To examine robustness of the data fusion approach in magnetically disturbed environments, the magnetometer data and therefore magnetic heading correction is turned off.

For eye gaze accuracy tests, a single calibration marker (AprilTag [92]) is presented to the user at five different locations to map the pupil positions onto the infrared camera's stream as described in section 6.2. After calibration, which takes about 30 s, the user randomly focuses different targets with a fixation time of around 1 s and without restrictions regarding head motion. The user presses and holds a mouse button upon fixating a target marker to trigger a record function that indicates the eye or head gaze is on the target. Upon button release, the trigger signal is turned off which stops the record function for this specific gaze target. The mouse button being pressed and released ensures that the user's intended eye or head gaze is on the target markers and thus rejects artificially introduced errors between target motions. The user is encouraged to move around with the chair in front of the table to evaluate robustness of the presented interface under dynamic motion conditions. A single trial takes 20 min. in total without the

calibration process.

The same setup is used to investigate head gaze accuracy. The laser diode is turned on to give the user feedback of the current head gaze point. Likewise to the eye gaze experiment, the user starts focusing targets with the laser dot for around 1 s without restrictions regarding head motion or position and toggles the mouse button when the target is in focus.

The data streams of the proposed interface (orientation, position, head/eye gaze point and on target event) and the ground truth motion capture data (orientation, position, target locations) are synchronized via timestamp-based filtering from a custom ROS node. The recording node ensures that maximum latency between the data streams is about 3 ms in total between the data streams. The data streams are synchronized at a 100 Hz rate.

## 6.4 Visual-inertial orientation estimation accuracy

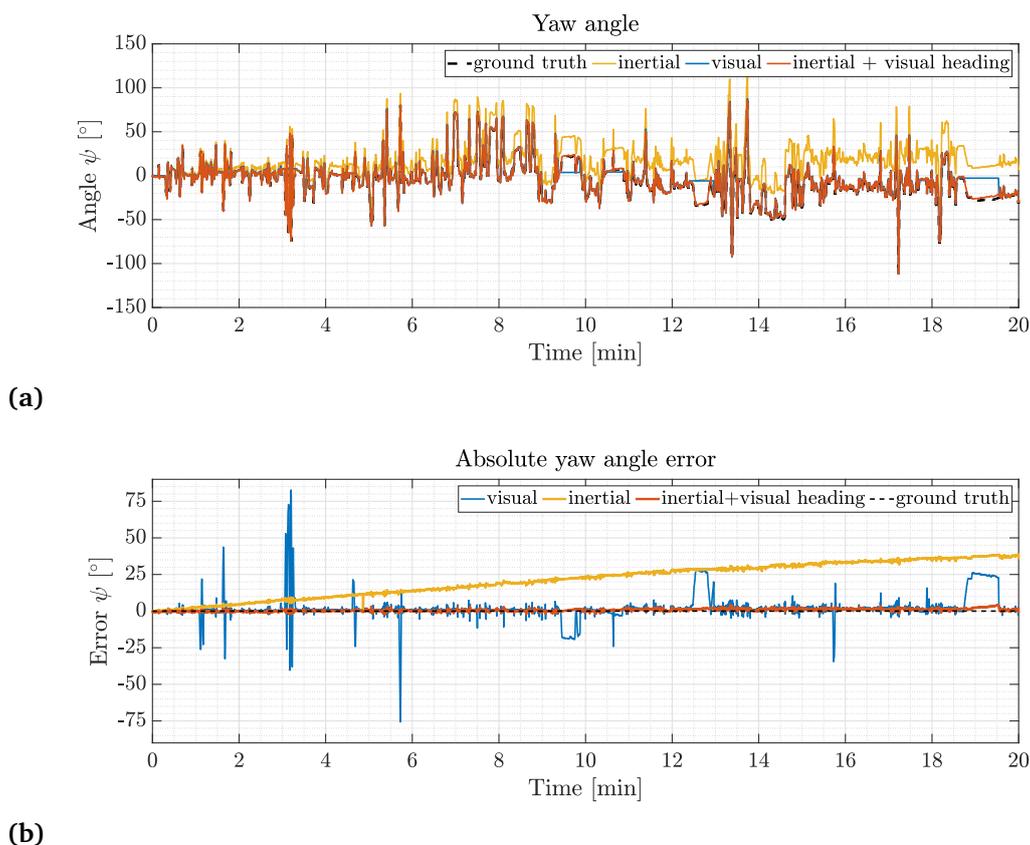
The orientation estimation accuracy is calculated as the mean of the RMSE values (root-mean-squared error) between ground truth Qualisys data, the visual-inertial and inertial-only orientation estimations for all 30 trials. RMSE values for visual-only data is not calculated since the RMSE will change with respect to the number of visual data losses and the magnitude of error compared to ground truth during these losses. Furthermore, the ORB SLAM framework is reset if relocalization fails within a time frame of five seconds and will be aligned with the MARG sensor's orientation estimation, compare section 6.1.2.

The results are presented as Euler angles in degrees, compare table 6.1. Throughout all trials, the visual-inertial yaw orientation estimation results in a mean RMSE of  $0.81^\circ \pm 0.44$  after 20 min. of continuous head motion. The inertial-only orientation estimation on the other hand results in a mean RMSE of  $12.49^\circ \pm 8.48$  for all trials recorded.

**Table 6.1:** Mean of RMSE values for inertial and the proposed visual-inertial orientation estimation [mean  $\pm$  standard deviation].

	Roll [°]	Pitch [°]	Yaw [°]
visual-inertial	$0.76 \pm 0.27$	$0.97 \pm 0.48$	$0.81 \pm 0.44$
inertial	$0.76 \pm 0.27$	$0.97 \pm 0.48$	$12.49 \pm 8.48$

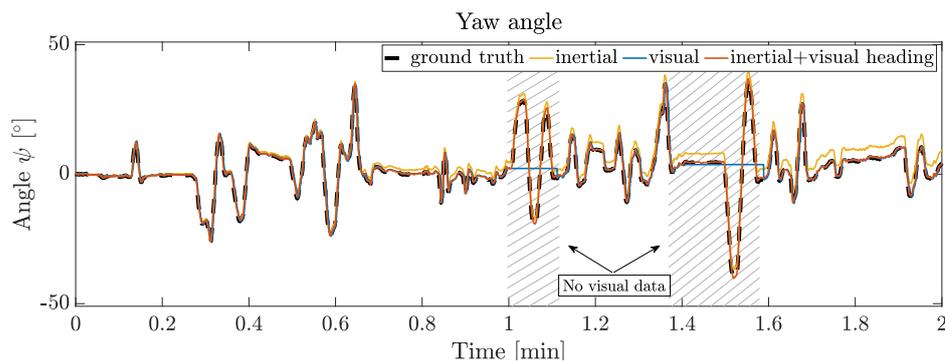
Figure 6.7a presents typical yaw angle results for the visual-inertial, inertial-only, visual-only and ground truth orientation estimation for a typical trial. The user randomly gazes at the 24 targets without restrictions to head motion. Figure 6.7b depicts the absolute error values from the subset. The maximum error for inertial only orientation estimation of this subset results in  $35^\circ$  accumulated drift after 20 min. whereas the visual-inertial orientation estimation results in a maximum error of  $3.7^\circ$  (compare minute 19.5) and



**Figure 6.7:** Yaw angle estimations and the corresponding absolute error for one trial: (a) yaw angle comparisons between ground truth (Qualisys, black), inertial only orientation estimation (yellow), vSLAM orientation estimation only (blue) and the proposed orientation estimation with visual heading vector substitute (orange). Figure (b) depicts the corresponding heading error referenced to the Qualisys system for either visual only (blue), inertial only (yellow) or visual and inertial yaw angle estimations (orange).

a total drift of  $0.5^\circ$  compared to the ground truth. While no visual data is available, the visual only yaw angle estimation experiences errors scaling with respect to the absolute orientation change, see, e.g., minute 1, 3, 9.5 and so on. Visual data loss originates from two different sources. Short period peaks of visual data loss are due to localization failure in the map. This occurs during dynamic motion in the visual scene between consecutive frames and the inability of the SLAM algorithm to relocalize with the current features in the given map. Longer visual data loss plateaus are due to intentional covering of the camera with the hand to prove the heading vector switching mechanism and robustness of orientation estimation during long visual occlusions.

Figure 6.8 depicts the heading vector switching mechanism during complete loss of visual data for a 6 s and 10 s period from an enlarged segment of the trial from figure 6.7a.



**Figure 6.8:** Sequence of yaw angle estimations during complete loss of visual data. Yaw Angle estimations: ground truth (black), the proposed orientation estimation (orange), the inertial only orientation estimation (yellow) and the visual orientation estimation only (blue). During complete loss of visual feedback (hatched areas) the filter switches the input heading source to the IMU heading vector to calculate reliable data until visual data is available again.

The proposed visual-inertial orientation estimation pipeline reduces the gyro-based drift accumulation for the yaw angle estimate to a minimum compared to the inertial only orientation estimation. During all trials the proposed fusion approach maintains the orientation and results in a mean RMSE drift of  $0.81^\circ$  in total whereas the inertial only heading estimate results in significant mean RMSE ( $12.47^\circ$ ) due to missing heading correction (see table 6.1). Pitch and roll angles are calculated based on accelerometer and gyroscope data and are typically less error-prone, since these angles are calculated based on the direction of gravity which at least can be measured in slow or static phases to correct for drift.

The proposed visual inertial orientation estimation pipeline significantly reduces accumulation of drift in the heading estimate in magnetically disturbed environments and enables full quaternion-based orientation calculation, even if the visual heading vector is not available, compare figure 6.8. During loss of visual data, the algorithm is capable of switching towards the IMU heading vector substitute to keep the orientation estimation stable. Even during long visual data losses of 40 s (see figure 6.7, minute 18.75), the proposed orientation estimation is able to produce reliable orientation data, even though it accumulates drift due to the gyroscope bias. As soon as visual orientation estimation is available again, the filter switches back to incorporate the visual heading vector into the orientation estimation pipeline. If relocalization in the map is successful, the accumulated gyroscope bias error can be subtracted, which is the case throughout the depicted trial. The proposed fusion scheme is able to switch to IMU only orientation estimation mode and maintain reliable heading estimation until visual feedback is available again. If magnetometer data is available and plausible (small relative deviation), the filter switches towards magnetic heading correction and this reduces error accumulation even further. Since the experiment is designed to evaluate the filter's robustness

while magnetic disturbance is present, the magnetic heading correction does not partake in the orientation estimation results depicted here.

## 6.5 Visual position estimation accuracy

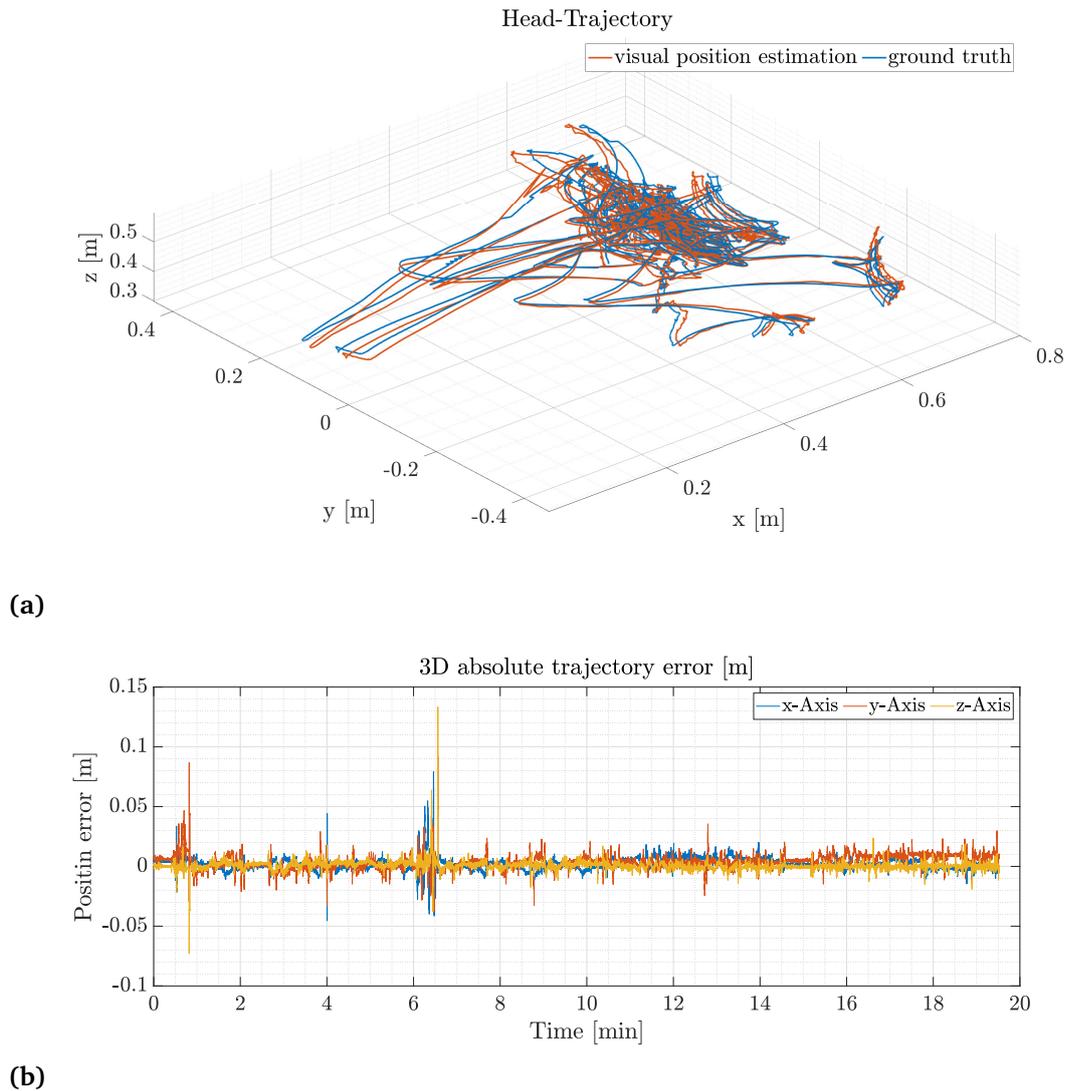
Trajectory estimation accuracy is calculated as the mean of the RMSE values of the visual position estimation and the ground truth Qualisys data and is presented in meters. The mean Euclidean position estimation error for the total of 30 trials is  $28.0 \pm 28.5$  mm, compare table 6.2. Figure 6.9a depicts a 3D representation of one subset of head trajectory measurements for the ground truth trajectory (blue) and the estimated trajectory of the ORB-SLAM visual position estimation (orange). In this trial, the user moves the head covering a total volume of  $0.75 \text{ m} \times 0.8 \text{ m} \times 0.2 \text{ m}$  (x, y, z) in total. Figure 6.9b depicts the absolute error for the subset in three individual axes, respectively. The maximum absolute error for an individual axis in this trial is 130 mm in the z-axis for a short duration of 0.2 s. The total RMSE for the depicted set is 9.1 mm.

**Table 6.2:** Mean of RMSE values for visual position estimation [mean  $\pm$  standard deviation].

	x [mm]	y [mm]	z [mm]	Total [mm]
Visual pos.	$16.8 \pm 18.6$	$20.8 \pm 21.0$	$8.2 \pm 5.2$	$28.0 \pm 28.5$

The trajectory estimation relies on visual position estimation from the ORB SLAM framework and transformation into the correct coordinate frame based on the visual and inertial based orientation estimation. This setup allows for accurate position estimation while visual pose estimation is available. If visual information vanishes, the last known position is held until visual position estimation is available again. If the local map from ORB-SLAM is sufficient, a relocalization and recovery is possible which will result in a small error in the position estimation. Relocalization is effective for example at minute 6.30 of the trial presented in figure 6.9b. The relocalization reduces the accumulated position error of the largest outlier from 0.13 m to 0.01 m in the z-Axis. If the track is lost, however, the map is reset which might introduce a position error that depends on the length of the visual feedback outage. The maximum length of visual outage without relocalization is limited to 3 s. Since the interface is worn by a human during robot teleoperation, the overall position change during a possible visual outage is rather limited and hence does not lead to large position errors. Furthermore, a map for a typical human-robot shared workspace is typically small which allows for fast mapping and small maps, which in turn helps with relocalization.

## 6.6 Three-dimensional gaze point estimation accuracy



**Figure 6.9:** Results for head trajectory estimations for one trial: (a) Ground truth head trajectory measured with the Qualisys system (blue) and the position estimation based on visual position estimation (orange). Figure (b) depicts the corresponding absolute trajectory differences for each individual axis between ground truth Qualisys data and the visual position estimation. The maximum single axis error is 130 mm in the z-axis.

## 6.6 Three-dimensional gaze point estimation accuracy

Gaze point accuracy is divided into two subsections, head and eye gaze accuracy, respectively. The accuracy for either method is calculated based on mean error values between ground truth values of the target points from Qualisys measurements and the estimated head or eye gaze point on the target. A mean gaze point is calculated for each of the 24 targets. A total of 30 trials was recorded, 15 for both head and eye gaze target positions.

### 6.6.1 Head gaze

Figure 6.10a depicts a typical subset to illustrate head gaze point accuracy for the x and y plane. The ground truth position for each target is presented as red circles whereas the head gaze points from this subset are depicted as blue circles. Every gaze target is focused multiple times during the trial, hence there are multiple gaze points (blue circles) for each target point. The head gaze trajectory for a whole target transition cycle between all 24 targets is depicted as a dotted black line. Several gaze points are distributed around the ground truth target point. The maximum head gaze position error for this trial is  $30.0 \pm 20.0$  mm in the x-axis,  $18.0 \pm 12.0$  mm in the y-axis and  $12.5 \pm 8.0$  mm in the z-axis. Figure 6.10b depicts the mean head gaze position error and its standard deviation for all 24 targets throughout all 15 trials. The mean Euclidean head gaze error for all trials results in of  $19.0 \pm 15.7$  mm, see table 6.3.

The head gaze error is increasing with distance between the participant and the target. Target points at the more distant end of the table have a higher standard deviation and larger mean error compared to the targets in the front of the table (see figure 6.10b). This is due to the scaling impact of orientation transformation errors, which have a high impact on the gaze point estimation at large distances with respect to the surface. The magnitude of error scales with the pitch angle relative to the surface and therefore a deviation of  $0.5^\circ$  at 0.3 m height results in a 1 mm error at a 0.1 m distance in the x-axis but results in 190 mm error at 1.0 m distance in the x-axis. The head gaze accuracy is furthermore reduced due to human errors when aiming for the targets. If the head gaze point (feedback laser) is not exactly coincident with the target midpoint, artificial errors are introduced that enlarge the standard deviation of the overall accuracy. Nevertheless, the head gaze accuracy is relatively high, see table 6.3.

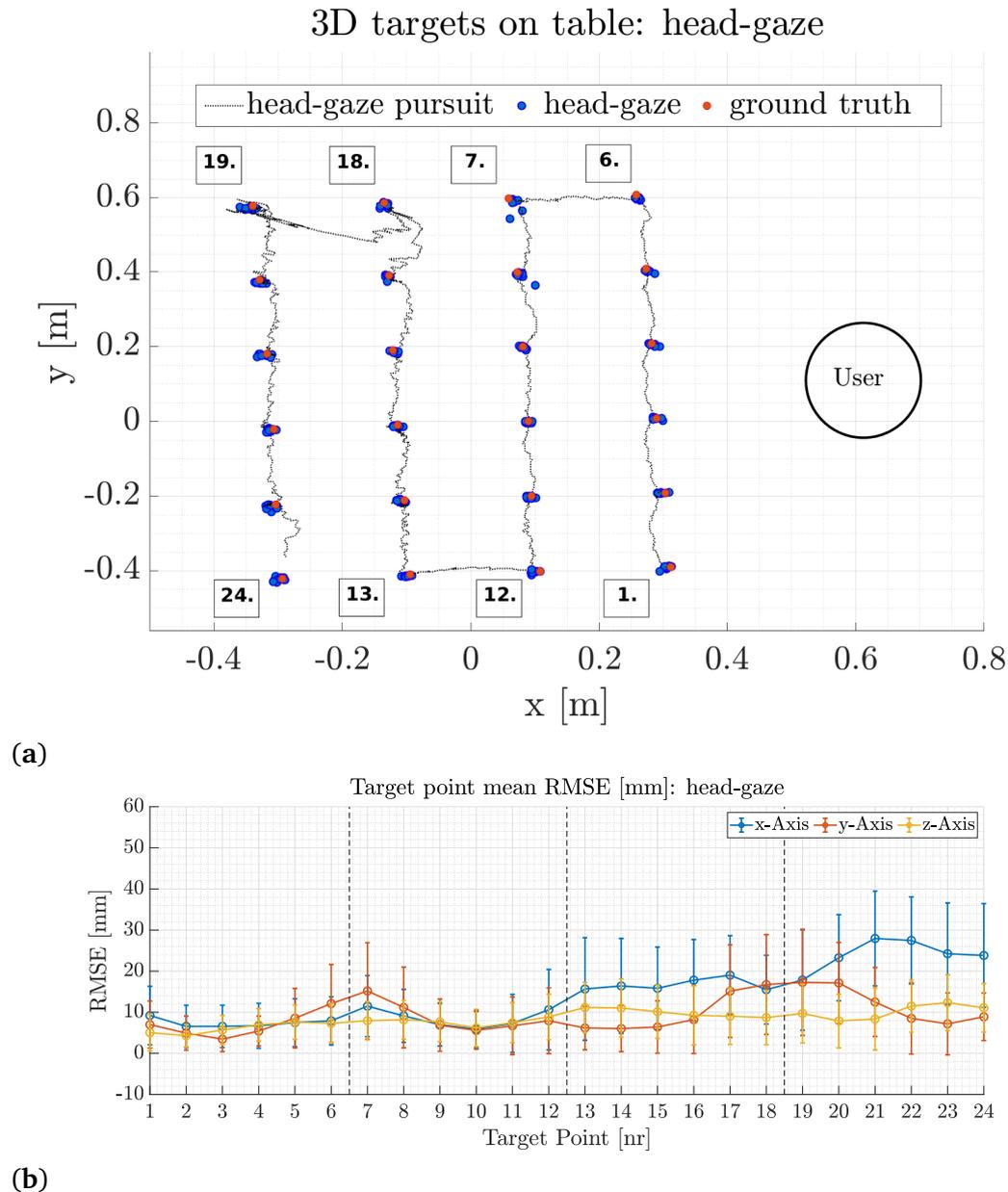
**Table 6.3:** Mean of RMSE values for gaze position estimations [mean  $\pm$  standard deviation].

	x [mm]	y [mm]	z [mm]	Total [mm]
Head gaze	$14.2 \pm 11.4$	$9.4 \pm 9.0$	$8.5 \pm 6.0$	$19.0 \pm 15.7$
Eye gaze	$17.7 \pm 12.3$	$15.4 \pm 12.7$	$14.2 \pm 12.7$	$27.4 \pm 21.8$

### 6.6.2 Eye gaze

Besides head gaze the interface and data fusion process presented here also enable three-dimensional eye gaze position estimation.

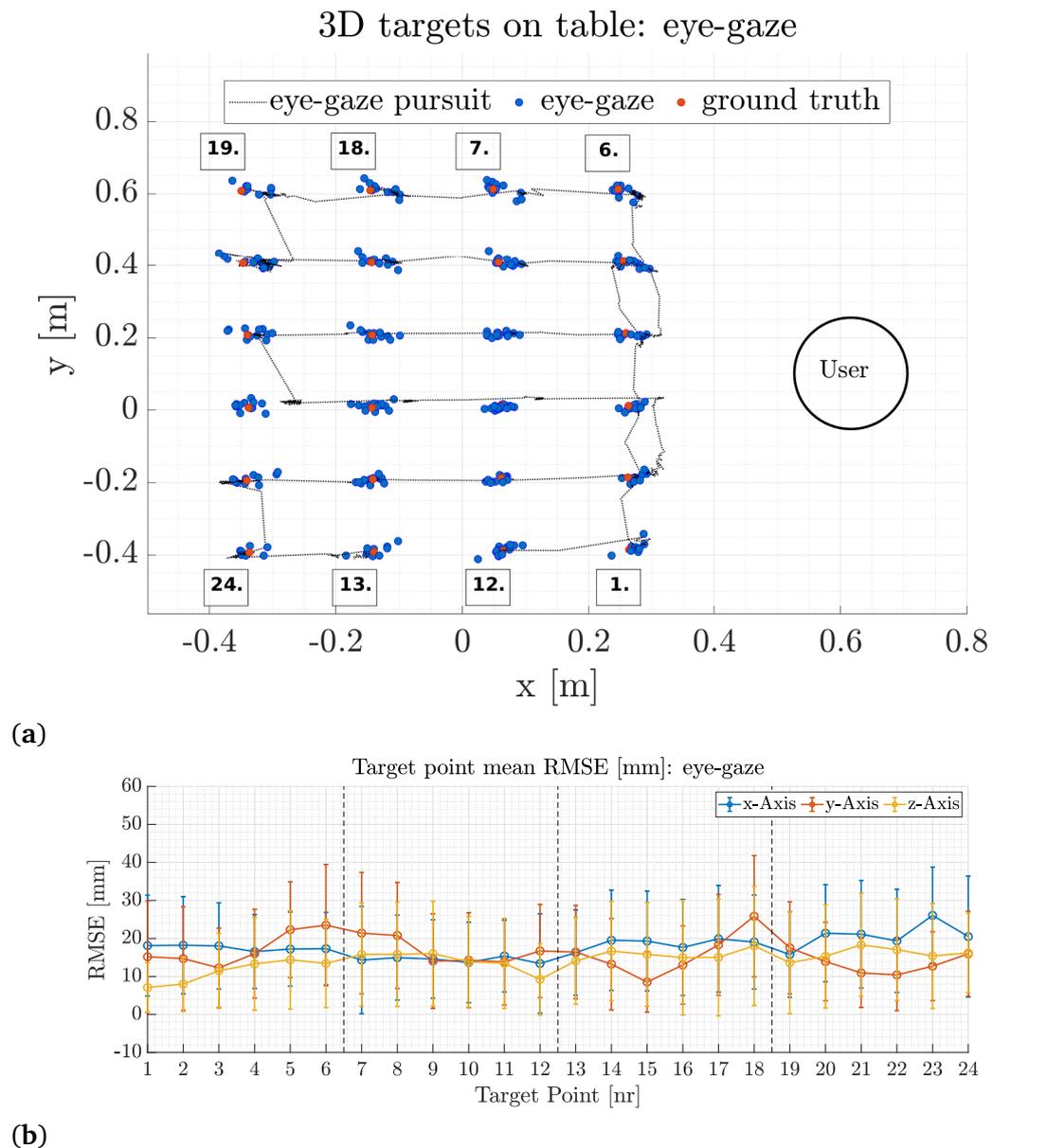
Figure 6.11a depicts a typical subset for eye gaze position estimation. The ground truth position for each target is presented as red circles whereas the mean target for the trial is depicted as blue circles. Every gaze target is focused multiple times during the trial, hence there are multiple gaze points (blue circles) for each target point. The eye gaze



**Figure 6.10:** Measurement results for head gaze accuracy: (a) Absolute position of head gaze on target position. The ground truth position of the targets (red circles) is based on prerecorded Qualisys measurements. The head gaze positions are depicted as blue circles. The head gaze trajectory (switching between targets) is marked with a dotted black line. The user's approximate position is represented as a black circle. Figure (b) depicts the mean head gaze error for each individual axis along target points throughout all trials. The black dashed lines indicate separation of the target groups for all four rows.

for one target transition cycle is depicted as a dotted black line. The largest eye gaze position error for the depicted trial is 50 mm in the x-axis for target point 23. This target is in the last row of the tabletop with the greatest distance to the user. The mean distance between these targets and the user is 1.1 m. The maximum y-axis error for this trial is 41 mm at target point 1. Likewise to the head gaze experiments, artificial errors are introduced from the user if the gaze point is not coincident on the target point, which in turn enlarges the standard deviation of the overall accuracy. Figure 6.11b depicts the mean eye gaze position errors and their standard deviation for all 24 targets throughout all trials. The mean Euclidean eye gaze error for all trials results in  $27.4 \pm 21.8$  mm, compare table 6.3.

eye gaze point estimation is less accurate when compared to head gaze point estimation, see Table 6.3. This is mainly due to eye gaze calibration inaccuracies which result in an offset or inconsistent map of the actual and calibrated gaze point. These inaccuracies in the gaze point estimation lead to the selection of a wrong depth pixel value which in turn results in a different 3D point in Cartesian space upon transformation. This effect does furthermore scale with respect to the distance between the camera's center and the target, compare section 6.6.1. The eye gaze error is increasing with distance from the participant and peaks in  $20.0 \pm 20.0$  mm maximum single axis error for points in the last row (1.1 m from head). Accurate gaze calibration is a prerequisite for 3D gaze point estimation. The eye gaze accuracy does also decrease due to slippage of the headset over time. The different calibration accuracies throughout the trials, slippage and human errors from target aiming will accumulate and result in higher standard deviations compared to head gaze accuracy, see table 6.3. Human control of a robot EFF in a small workspace is enabled by eye gaze as well as head gaze point estimation. head gaze control could be preferred in a larger workspace and for a more precise control approach. Comparing the presented interface to a recent 3D eye gaze interface proposed in [30] demonstrates the presented interface's higher accuracy in terms of total Euclidean error (27.4 mm vs. 46.8 mm) but has a lower repeatability (21.8 mm vs. 1.4 mm). This effect is mainly due to the prior mentioned inaccuracies from eye gaze calibration, slippage and furthermore position tracking accuracy differences. Shaftie et al. use an infrared-based motion capture system which gives significantly higher resolution compared to the vSLAM based position estimation. This can also be seen in table 6.2, since an infrared motion capture system is used as ground truth to benchmark the visual position estimation.



**Figure 6.11:** Measurement results for eye gaze accuracy: (a) Absolute position of head gaze on target position. The ground truth position of the targets (red circles) is based on prerecorded Qualisys measurements. Absolute eye gaze positions on a target are depicted as blue circles. A sequence of one complete transition between targets through eye gaze is depicted as a dotted black line. The user's approximate position is represented as a black circle. Figure (b) depicts the mean error of the eye gaze for each individual axis along target points throughout all trials. The black dashed lines indicate separation of the target groups for all four rows.

## 6.7 Conclusion

This work presents a mobile head-worn interface that enables a user to accurately control a robotic arm in 3D Cartesian space via head or eye gaze. Furthermore, it enables to control the orientation of the end effector of a robotic arm by using accurate 3D head motion angles. The exclusive use of a camera with vSLAM method or the exclusive use of MARG sensors for robust estimation of the heading angle shows different intrinsic features in the time domain. While purely visual data from the vSLAM method has disadvantages at higher dynamics, i.e., at strong relative movement of the robot arm in the scene, the IMU sensors show a drift of the heading angle at very slow movements of the head. This work estimates a robust heading angle by fusion of data from a vSLAM camera with that of a MARG sensor. Even though other interfaces exist, these might be obstructive or too heavy for people with severe physical disabilities, e.g., people suffering from paraplegia. The proposed interface is lightweight and mobile and can be used without the need for external reference markers or static cameras.

The data fusion process generates robust and accurate orientation and position estimation of a user's head with respect to a dedicated workspace in indoor environments, and is capable of switching between visual-inertial, inertial only and inertial-magnetic orientation estimation, based on reliability of sensor data. It is furthermore possible to utilize the SteadEYE-Head approach presented in chapter 5. The visual fixation detection and sample and hold mechanism for the heading vector can be seamlessly integrated into the approach. The presented data fusion is infrastructureless and therefore not dependent on any external references, e.g. fiducial markers, stationary camera equipment, and so on. The data fusion process is capable of delivering robust orientation and position estimation, even while subject to significant relative motion from the robotic system or fast-changing magnetic disturbances.

The possibility of using head or eye gaze control enables an intuitive communication channel for robot collaboration that facilitates natural gaze-based task interaction. Depending on the desired accuracy and size of the workspace, head or eye gaze could be used interchangeably. For example, the 3D eye gaze point could be used to quickly determine a ROI for the robot and switch towards head gaze for accurate control of the end effector position. If the eye gaze experiences an offset due to slippage from the eye tracker frame, a user can switch towards head gaze mode and thus maintain control and safety. To further enhance robot end effector positioning precision a mixed control strategy could be used.

## 7 Verification in a human-robot collaboration use case

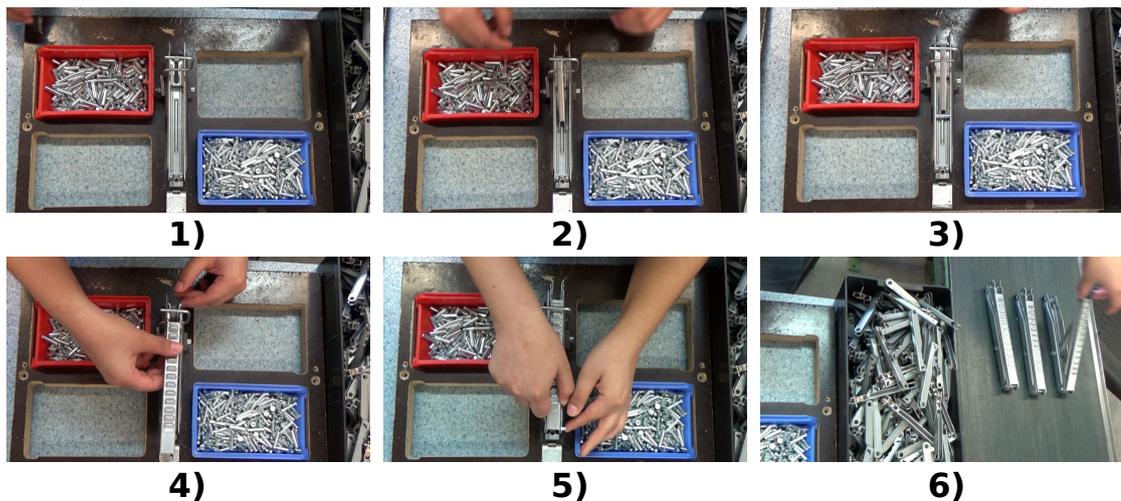
The following chapter presents the use case that was chosen to evaluate the proposed methods in a real-world use case. First, the chosen assembly task is presented, followed by the derived workplace setup. Afterwards, the control interface and designed control strategy are explained in greater detail. Finally, the experimental results for a preliminary Cartesian point accuracy test as well as the functional test of the developed task are reported.

### 7.1 Use case – mechanical assembly of wedge adjuster

To verify the functional usability of the developed methods and the interface for precise and safe human robot collaboration a real-world assembly task was chosen. It was chosen based on a preliminary visit and study of possible human-robot collaboration tasks at Büngern Technik, a workshop that employs physically and mentally disabled people. The goal of this study is to find and define a mechanical assembly task and transform this into a human-robot workspace to enable a physically disabled person to solve the assembly task through the developed interface and robotic teleoperation respectively.

The task itself consists of a mechanical assembly of a "Keilsteller" (wedge adjuster). A Keilsteller is used in slatted frames of beds to adjust the inclination of the top part of the frame. The task is usually executed from able-bodied workers. A worker is in need of both hands to assemble a full Keilsteller and furthermore utilizes an assembly aid to hold the different parts in place during the assembly. The wedge assembly task can generally be simplified and separated into two iteratively occurring tasks: a) the isolation and grasping of a part of the assembly group from bulk containers and b) the placement of the part in an assembly aid or assembly frame. This sequence is repeated five times for a total of five components, which form the assembly group.

The complexity of the prior mentioned procedure varies with respect to the geometry and size of the component, e.g. separating and placing a rivet in contrast to a larger component which is easier to grasp. Both parts of every sequence are in need of high precision while only the separation task furthermore demands decision-making skills to enable a successful isolation of a single part from the bulk container. This difference contemplates a discrete separation into human- and robot-centered tasks, based on their



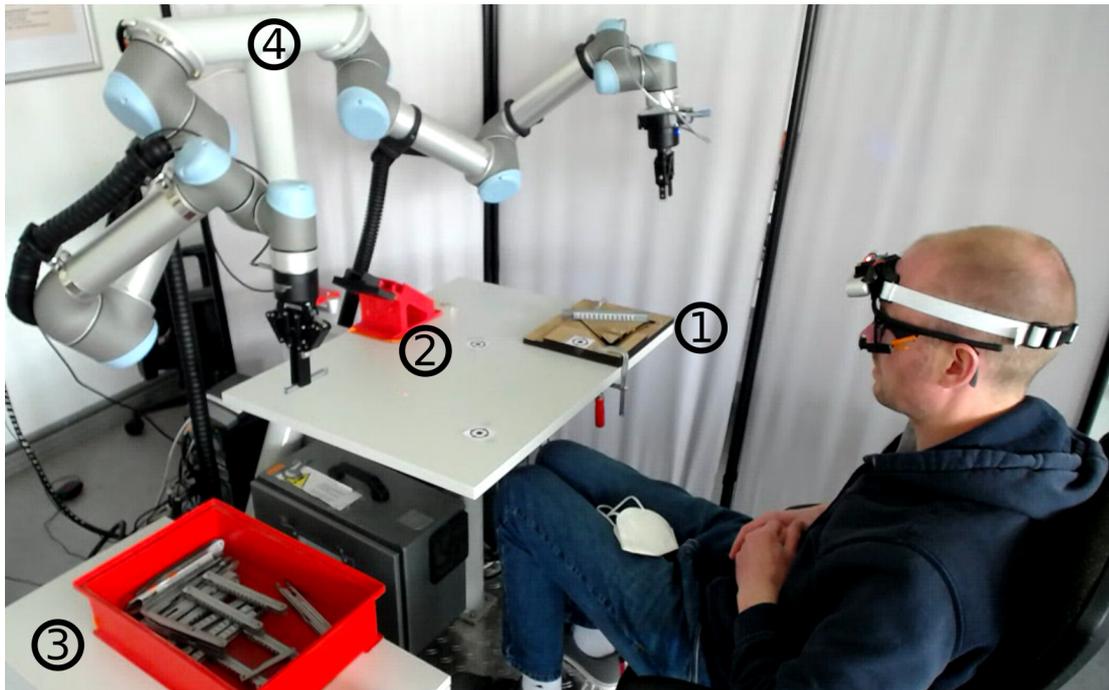
**Figure 7.1:** Full wedge adjuster assembly process (Keilsteller). 1 – Bottom part placed on assembly frame, 2 – lift arm placed on hanger (assembly frame) and aligned with center hole from bottom part, 3 – insertion of rivet between lift arm and bottom part, 4 – top part slid over lift arm and aligned with rear end hole, 5 – insertion of rivet between top and bottom part, 6 – finished wedge adjuster (Keilsteller) placed on conveyor belt.

respective strength and weaknesses. Humans do have a superior visual system coupled with direct decision-making skills based on past experiences. A human is able to identify the orientations and positions of a desired component in a bulk container in no time. This information can be used by the human to control a robotic system (a robotic arm) for the separation and grasping of a component. Since the components are not rotationally symmetric, it is difficult to automate this process, e.g. using computer vision and machine learning approaches to track and grab a single part.

The robotic system, in turn, can provide the precision and repeatability needed to assemble the components into a functional group. Especially the placement of the rivets within the assembly group needs high-precision grasping and placing of the part. Thus, it is ideal to fully automate this assembly process and give the human operator the ability to intervene and correct the robot pose if needed.

## 7.2 Workplace setup

Based on this division of tasks, a workplace was configured that enables the assembly process and the aforementioned division into human- and robot-centered subtasks. The robotic system consists of a dual arm UR-5 mounted at an angle of  $45^\circ$  onto a t-beam. Furthermore, a tabletop is welded to the t-beam that represents the robot's workspace. Both robotic arms are connected via an RS-485 interface to their cabinet controllers. The cabinet controller is connected via Ethernet to a local network. A dedicated computer



**Figure 7.2:** Dual arm robotic workspace. Overview of the robot workspace setup: A user is sitting in front of the dual-arm UR5 robotic system (connected via the t-beam ④, the frame ① and magazine are fixated on ② on the tabletop, while the bulk container ③ is placed on a counter left to the robot.

is connected to the local network running the ROS core, functioning as the server for a multiple machine ROS network.

This master server is in charge of the trajectory generation and communication with the UR-5 robot cabinets. A control application at the top-level listens for incoming messages on a dedicated control message topic. Any ROS-enabled computer might connect to the master server and publish commands onto the desired topic. This setup enables the use of multiple machines of varying complexity or hardware. Both arms are equipped with a Robotiq 2F-85 gripper that is plugged into a USB multiplexer that enables ROS based gripper control or cabinet gripper control. The latter is used to call a preprogrammed assembly script from the cabinet instead of using ROS. The USB multiplexer can be switched from ROS to cabinet control from a simple mechanical switching apparatus that is controlled using an Arduino microcontroller that has a dedicated interface to the robot application on the ROS computer. Figure 7.2 depicts the workplace setup.

The human teleoperates one of the two robot arms to a bulk container via the interface described in chapter 6 and separates a component. This component is then placed at a specific slot in an assembly magazine. The only thing that is of imminent importance is that the part is placed in a specified orientation and in the correct slot. The second robot arm can grab the component from this slot in the magazine and insert it directly

into an assembly frame. Since both the magazine and assembly frame constrain the part position and pose, the assembly process can be fully automated. The magazine as well as the assembly frame were specifically designed as part of the workplace. The assembly frame was constructed to optimally utilize the components' geometries and takes advantage of directed use of forces from gravity as well as magnets placed under the positions for the rivet holes.

### 7.2.1 Control strategy

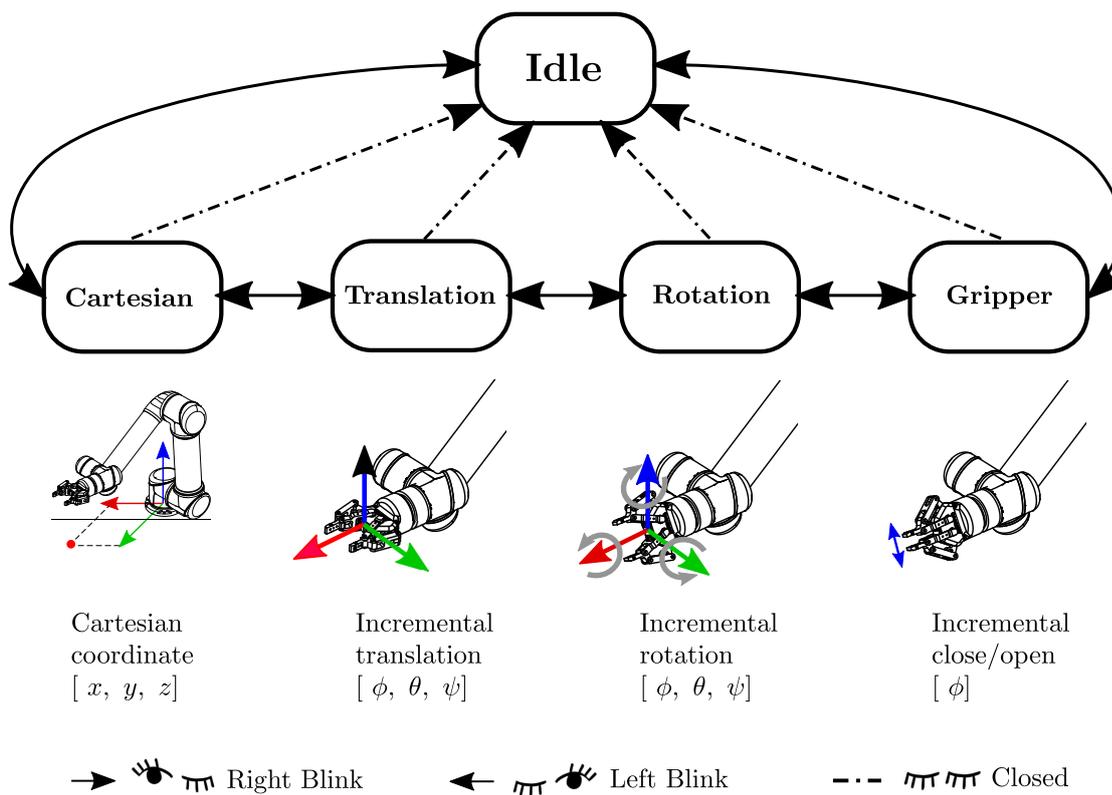
There are various strategies for control of a robot in 3D space by head motions. In this thesis two control strategies are combined to precisely control the robot's EEF in Cartesian space in all 6-DoFs. The first control signal uses head or eye gaze to rapidly steer the robot's EEF to a desired point in 3D Cartesian space, while the orientation stays fixed. The second control strategy utilizes head motion to incrementally change the robot's EEF orientation or position. This strategy employs a motion mapping between the 3-DoF of the human's head rotation onto the robot's EEF orientation or position. This is based on the work from Rudigkeit et al. [21, 16]. The interface introduces 5 different control groups that are organized in a circular buffer fashion. Switching from one to another group is based on iterating an internal counter by means of left or right eye blinking for more than 0.7 s. Figure 7.3 illustrates the motion control groups and control signals.

**Idle** The starting point is the idle state. In this state, the robot motion control is blocked, giving the user the ability to freely move around. Every time the user closes both eyes for more than 0.7 s the idle group is activated, immediately discarding all control signals.

**Cartesian** In Cartesian control mode, the robot can be commanded to any 3D  $(x, y, z)$  point inside the robot's workspace by means of head or eye gaze. This can be implemented in both a continuous control (following the gaze point) or by approving the point through a visual fixation for more than 0.7 s.

**Translation** In the translation control mode, the robot's EEF position in  $x, y, z$  can be incrementally increased by a proper head rotation  $(\phi, \theta, \psi)$ . Upon changing into this motion group, the current head orientation is sampled and set as the zero-orientation baseline for the dead-zone threshold. The threshold is empirically set to be  $\pm 15^\circ$  in any direction, that has to be exceeded before the robot control is enabled.

**Rotation** In the rotation control mode, the robot's EEF orientation can be incrementally increased by a proper head rotation  $(\phi, \theta, \psi)$ . Likewise to the translation mode, the current head orientation is sampled and set as the zero-orientation baseline for the



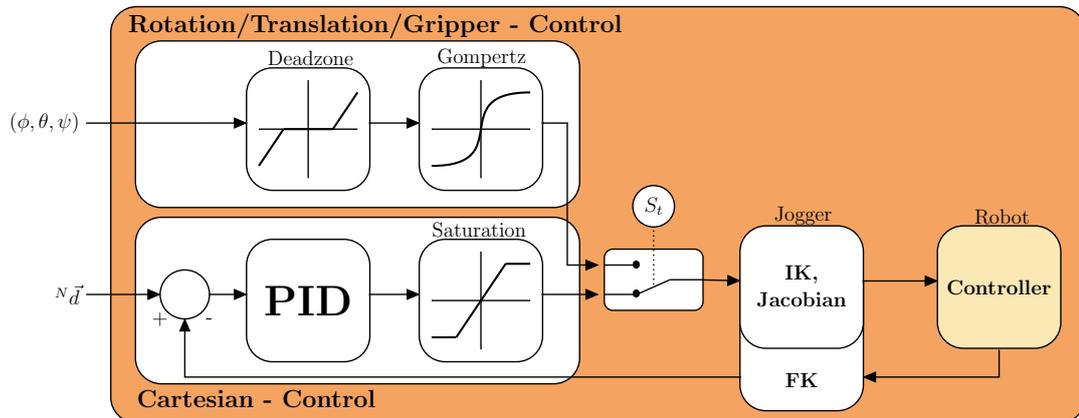
**Figure 7.3:** Motion control groups for Cartesian robot control. Eye blink states are used to switch between control groups, while 3D gaze-point and head-motion angles are used to teleoperate the robot in the chosen control group.

dead-zone threshold upon entering the group. The threshold is empirically set to be  $\pm 15^\circ$  in any direction, that has to be exceeded before the robot control is enabled.

**Gripper** In the gripper control mode, the two-finger gripper can be incrementally closed or opened based on a positive or negative pitch rotation of the head ( $\phi$ ). Again, the current head orientation is sampled and set as the zero-orientation baseline for the dead-zone threshold upon entering the group. The threshold is empirically set to be  $\pm 15^\circ$ .

The user can switch towards the appropriate control mode by a simple eye blink which toggles the state variable. The controller design for either control group is depicted in see figure 7.4.

In position control the 3D gaze point ( ${}^N\vec{d}$ ) is fed to closed-loop Proportional Integral Derivative (PID) controller anti-windup and saturation. The error term is calculated from the desired 3D point (head or eye gaze) and the robot's EEF position, which is calculated based on the Inverse Kinematics (IK) equation. The output is a Cartesian velocity vector which is saturated to enforce speed limitations onto the robot for increased safety. This vector is fed to a velocity-based Cartesian jogger. The jogger calculates the



**Figure 7.4:** 6-DoF robot control block diagram. The trigger signal  $S_t$  toggles Cartesian position, translation, orientation, and gripper control. The toggle is set when an eye blink or other discrete events occurs.

desired joint positions and velocities and publishes both to the robot’s ROS interface which communicates with the robot’s internal controller. The robot’s joint angles are used to calculate the EEF’s new pose through its Forward Kinematics (FK). The new pose is looped back to calculate the new error term. This simple PID jogging allows for smooth and continuous 3D position robot control in Cartesian space. During jogging, the robot’s EEF orientation is held constant since orientation is not represented by the 3D head or eye gaze point. The Cartesian position control is generated through a closed-loop PID controller with anti-windup and saturation.

In translation, orientation or gripper control the pitch-, roll- and yaw-angles given by the user’s head motion are mapped to the 3D orientation of the robot’s EEF. This motion mapping has been intensively studied and allows for precise incremental control of a robot’s EEF orientation and position [21, 16]. Switching from position- to orientation-control and vice versa is based on blinking with the right eye. Upon changing and entering a head-angle based control group (either translation, rotation or gripper control), the current orientation of the head is sampled and used as zero-orientation baseline with a  $15^\circ$  dead-zone (see figure 7.4). The user needs to rotate the head to the desired angle (pitch, roll, yaw) beyond the dead-zone to rotate the EEF. The angular or linear velocity of the EEF’s position- or orientation-change scales with respect to the relative head-angle change. A bigger angle equals a higher linear/angular velocity, whereas a small angle results in low linear/angular velocity. The orientation of the EEF is mapped based on a Gompertz function if the head angles exceed a certain dead-zone. Zeroing the head position at every motion group change has a simple but rigorous effect regarding the mapping and recalibration needed in [26]: In the original implementation, even a tiny accumulation of drift over time would lead to a deviation of the initial zero-orientation, which is used to define the dead-zone. The small drift shifts the initial pose that has to

be maintained in order to prevent the robot from moving. This applies to any further accumulation of angle errors and in the worst case leads to a complete misinterpretation of the control signals. To solve this problem, the initial position must be recalibrated. This recalibration has to be triggered, in case of a pure IMU interface control and switching signals are triggered from the same source, posing a problem since the switching signal will also be subject to the drift. The multimodal interface can provide a remedy here, since in this case eye blink or alternatively eye gestures trigger the recalibration (switching signal), immensely simplifying recalibration of the zero-orientation and re-enabling robot control.

Either the saturated position or angle increment is fed to the jogger that calculates the IK and Jacobian to iteratively increment the robot's position or orientation while the input is non-zero. This setup allows for full continuous Cartesian motion control for position and orientation of the robotic system.

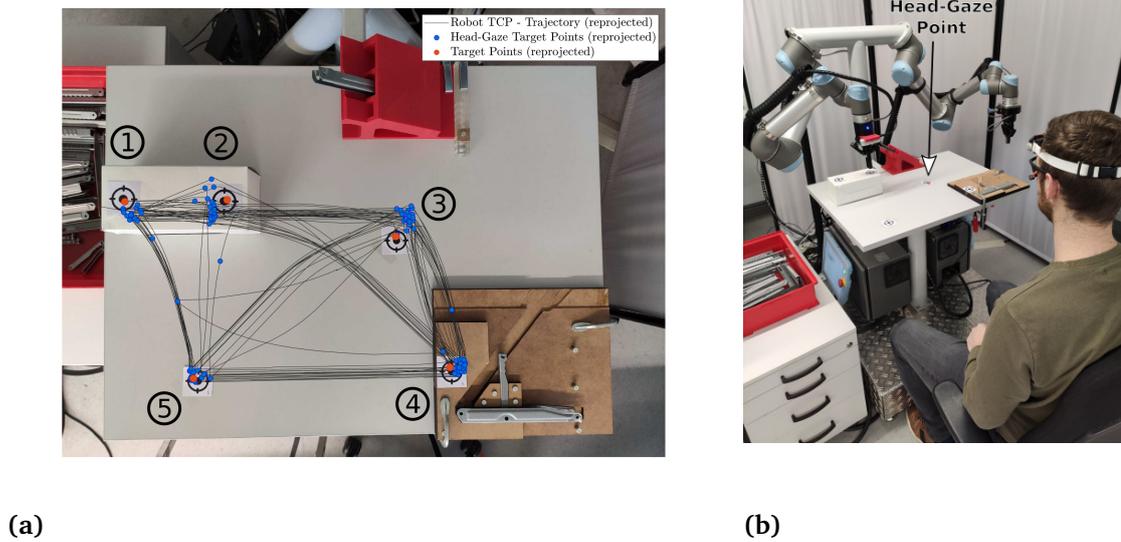
### 7.3 Results

A proof of functionality of the interface for real-world gaze-based robot control was conducted using the interface alongside the developed robotic control scheme presented in the previous section. Since the goal of this thesis is the development of a robust infrastructureless sensor system for human-robot collaboration rather than the development of a new control strategy, the results do not include task load tests or any other usability measure. The control paradigm that is used and adapted here was already tested regarding task load and usability in [26]. The results obtained here are focused on the accuracy of the head and eye gaze point and therefore on the head pose accuracy, since the gaze point is directly derived from this quantity.

The user randomly gazes at five different target points inside a robot's working area for 20 min. in total. The user blinks with the left eye to send the gaze point to the robot control pipeline upon which the robot moves to this point. The user sits in front of a dual arm robotic system and is in control of one arm, see section 7.2.1 for more details. The target positions are placed at three different heights (14 cm, 2 cm, and 0 cm with respect to the table surface) to demonstrate three-dimensional position control. The target positions are known with respect to the robot coordinate system and are compared to the robot's tool center point (TCP) position.

#### 7.3.1 Control accuracy of Cartesian pointing

Figure 7.5b depicts the workspace and the gaze targets for the robot control. The robotic system consists of a dual arm UR5 mounted at an angle of  $45^\circ$  onto a T-beam. Both arms are equipped with a Robotiq 2F-85 gripper. Furthermore, a tabletop is welded to the T-beam, which represents the robot's workspace. The robot control application provides a complete implementation to control each arm separately or even control both arms



**Figure 7.5:** 3D gaze point robot control application. (a) depicts the re-projection of the gaze points (blue circles), target points (red circles) and the robots TCP (black line) onto the actual workspace image. (b) shows the robot workspace application. A user is sitting in front of the dual arm UR5 robotic system. The user aims for the targets using the head gaze approach. Upon an eye blink (left eye), the gaze-point is transferred to the robot control pipeline.

simultaneously. For this proof of functionality, however, the user only controls a single arm (left side).

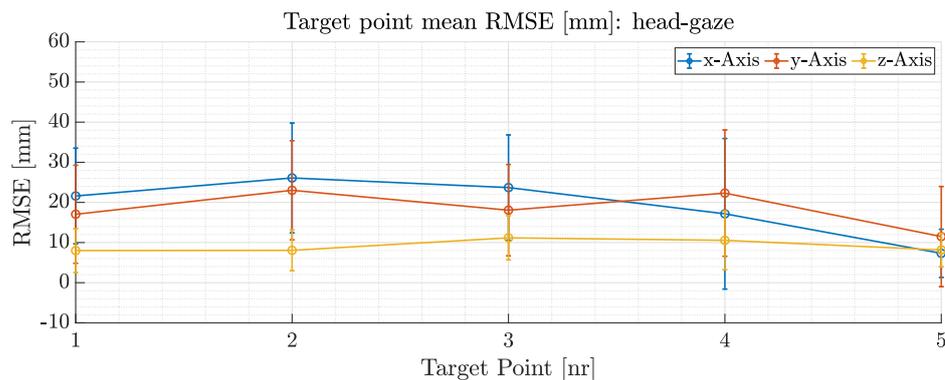
The motion parameters of the robotic system are limited to ensure a teleoperation that is safe for the human. The parameters are listed in table 7.1. The experiment does not involve any direct human-robot contact. This is enforced through the physical distance between the user and the robotic arm which is larger than the maximum stretch limit of the robotic arm in the humans' direction (1.1 m including the gripper). The user aims at five different waypoints on the tabletop in the workspace. Upon a discrete event (blinking with the left eye) the Cartesian gaze point is transferred to the robot control pipeline of figure 7.4, actuating the robotic arm.

Figure 7.5a depicts a top view re-projection of one trial for ground truth target positions (red dots), the commanded head gaze points (blue dots) and the robot's 3D tool center

**Table 7.1:** Motion parameters for the UR5 teleoperation (vel. = velocity, acc. = acceleration).

Linear vel. [mm/s]	Rotational vel. [rad/s]	Linear acc. [mm/s <sup>2</sup> ]	Joint vel. [rad/s]	Joint acc. [rad/s <sup>2</sup> ]
50	0.4	100	0.4	0.7

### 7.3 Results



**Figure 7.6:** Mean RMSE error of the head gaze for each individual axis along target points throughout five trials during robotic teleoperation.

point trajectory (black line) on the robotic workspace. The mean RMSE head gaze values and the standard deviation for the five target points from a total of five trials are plotted in figure 7.6.

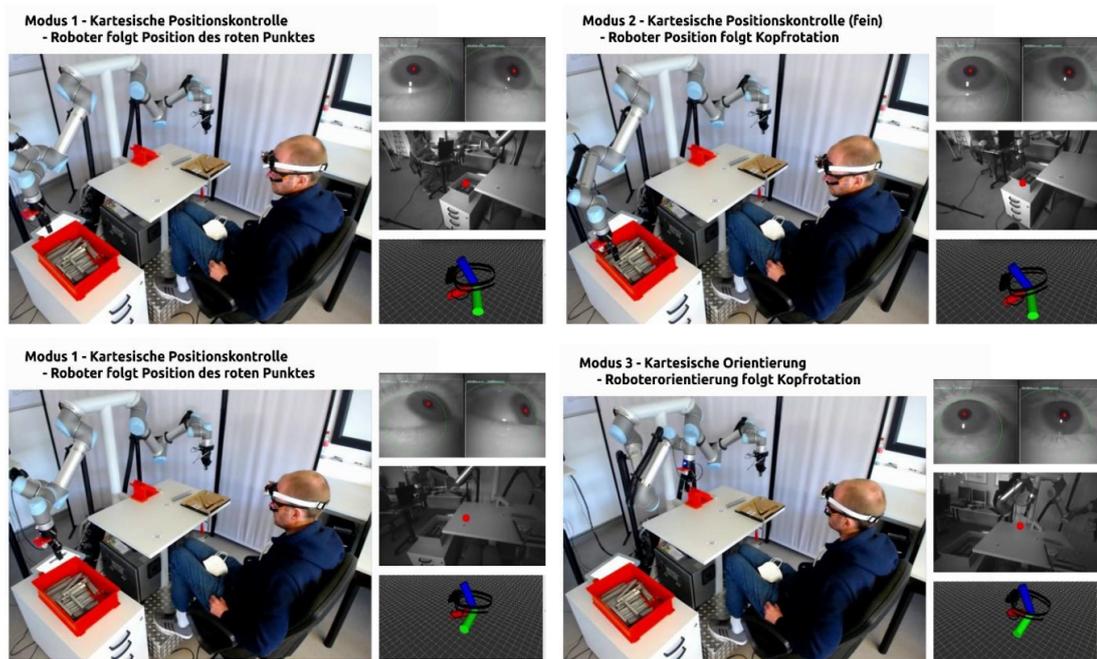
The mean Euclidean RMSE for head gaze-based position control results in  $26.5 \pm 20.9$  mm for all five target points inside the test workspace. The total mean RMSE for all five targets is only 7 mm larger compared to the 24 targets' head gaze total mean RMSE that only had very little relative motion in the visual scene. This demonstrates the overall usability of the proposed interface and methods for gaze-based robot control even while the interface experiences relative motion of the robotic arm in the visual scene and magnetic interferences being present.

Using three-dimensional gaze points expands the direct robot teleoperation approach beyond incremental position change. Since the gaze signal also decodes depth information of the point on the surface, the robot trajectory can be constrained to ensure a safe robot motion. Consider the following use case: The desired gaze point is below the surface point of the table.

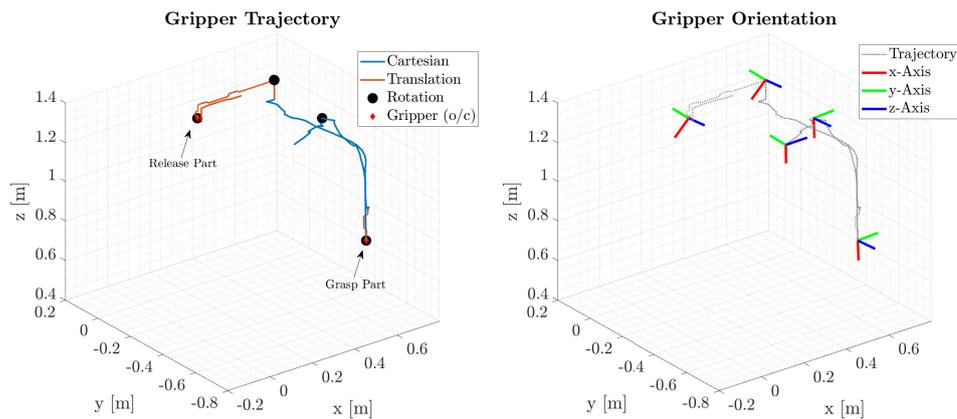
### 7.3.2 Teleoperation – isolation and grasping

In the following, the results for the isolation and deposition task of components from bulk containers will be presented. Due to the fact that within this thesis the development of the multimodal sensor system as well as the methods for robust control variables computation are the main research topic, there is no extensive user study concerning the usability of the control architecture or the task load. The results presented here serve to illustrate the functionality of the sensor system and the designed methods for the targeted isolation and precise positioning of components via teleoperation of a robotic system. Since the control architecture is based on [26] and partially on [20], the reader is referred to the aforementioned papers for in-depth analyses of the general usability and taskload investigations with healthy participants as well as people suffering from tetraplegia.

The functional validation of the interface for precise robot teleoperation is acquired using the workplace and assembly task presented in section 7.2. A user is instructed to isolate and grasp a single part from a bulk container, translate it to the magazine and



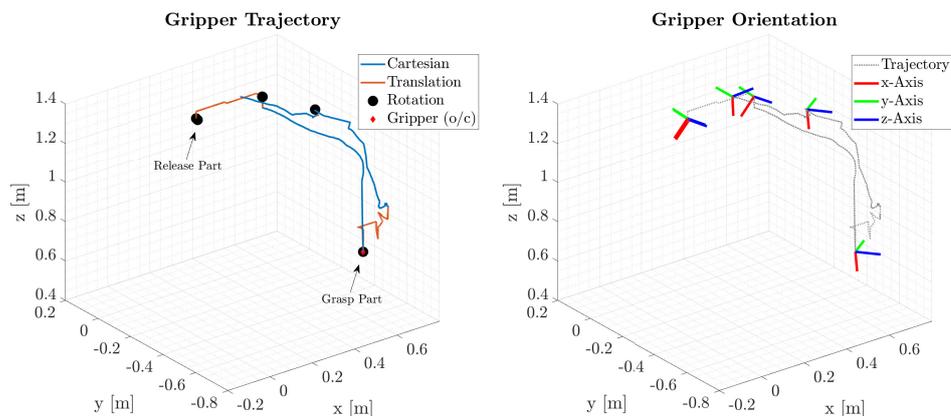
**Figure 7.7:** Image sequence of robot application in use. The user gazes at the bulk container (head gaze – red dot in black and white image) to translate the robot to the desired point. Next the user corrects the position and orientation via motion mapping and grabs the part. The user gazes at the desired location of the object and the robot translates the grabbed part to this point. Finally, the user rotates the part to match the orientation of the magazine to release the part into the correct slot.



**Figure 7.8:** EEF trajectory and orientation depiction from a single isolation and grasping trial. The left plot depicts the proportion of each motion group forming the total trajectory of one isolation and grasping trial of a single part. Black dots indicate phases where the gripper is rotated. The right plot depicts the same trajectory but emphasizes on the orientation change of the gripper at the end of each rotation phase (black dots from left plot).

release the part into the correct slot. The user can freely teleoperate the robotic arm without any restrictions regarding approach of the bulk container or magazine of any kind. The user can freely switch between the motion groups by a simply left or right eye blinking.

Figure 7.8 and 7.9 depict the robot's TCP with respect to the common world coordinate system. The complete trajectory for a single part separation and placement in the magazine is subdivided into their retrospective motion group (color coding). It can be seen that the user does utilize the Cartesian point-to-point motion to cope with the larger



**Figure 7.9:** EEF trajectory and orientation depiction of a second single isolation and grasping trial.

distances between the magazine and the bulk container, since it does only require the user to gaze at the desired point of regard. This is expected, since the incremental translation approach would require multiple head-angle changes with long periods of simply holding the angle in the desired direction, while it is more convenient to just gaze at the desired point. In contrast, the iterative translation approach is used for fine adjustments of the EEF to enable grasping of a component or when inserting it precisely into the magazine. The mean execution time for a single trial (separating, grasping, and placing a single component) is 310.87 *s* with a tolerance of just 0.4 *cm* per slot.

The proof of functionality contributes to the overall conclusion, that the interface alongside the developed methods enable a precise robot teleoperation and can be used for human-robot collaboration tasks.

## 8 Discussion

The algorithms developed and tested enable an improved orientation estimation when magnetic disturbance is present without the need for additional infrastructure, limiting the possible use case or motion space. The functional combination of MARG sensor and head-worn eye tracking data enhances the orientation estimation for low-cost consumer-grade MARG sensors. The methods developed here, can be used regardless of the hardware. The algorithms can benefit future head-worn technologies by means of using eye tracking feedback for head orientation estimations for indoor environments.

Chapter 4 presented a robust linear Kalman filter employing a gradient descent method for robust orientation estimation and short-time magnetic disturbance compensation. Adopting the gradient-descent method to compute a measurement quaternion for a linear Kalman filter does simplify the implementation and computational cost and therefore contributes to a real-time self-contained sensor system. Calculating a full quaternion prior to the data fusion with the gyroscope enables the possible usage of other attitude and heading references (compare chapters 5 and 6). Thus, the data fusion process contributes to the modularization and functional combination of different sensing technologies to enable robust and infrastructureless orientation estimation. Using the unit quaternion as the measurement and system state reduces the overall matrix dimensions, leading to a reduced computational load on the sensor system itself. The magnetic disturbance detection enables a switching mechanism to avoid and overcome short-time magnetic disturbances by switching towards an IMU-type set of equations. In contrast to magnetic norm-based disturbance detection, the presented approach can be used with any heading information and does not rely on a specific field value. In fact, it only relies on the relative differences between the quaternions from MARG and IMU stage. The developed framework is able to compete with currently existing MARG sensor filter frameworks and even outperforms a commercially available MARG sensor that was used in a related work of N. Rudigkeit for the development of head motion-based robot control. Nevertheless, the system is only capable of overcoming short-time magnetic disturbances because it will accumulate heading errors stemming from the DC bias of the gyroscope over longer periods of time in the IMU-mode. Besides this, the filter framework in its current form lacks a DC bias estimation method to further decrease the accumulation error and prolong robustness during magnetic disturbances.

To increase robustness and ensure safety of the human collaborator during long-time magnetic disturbances, chapter 5 presents a novel method for infrastructureless orientation estimation that exploits eye tracking data to support MARG sensor-based head

orientation estimation. The approach takes advantage of the physical connection between head- and eye-rotation during visual fixations of objects, that can be measured if a MARG sensor and eye tracker are both mounted in fixed position on the head-worn frame. In conclusion, a MARG or IMU sensor data fusion pipeline samples and holds the current orientation, isolates the heading component, and recursively feeds it to the input stage while a visual fixation is detected. A visual fixation corresponds to a near-zero or marginal pupil position change in the eye camera's image over a specified period of time. This mechanism is somewhat similar to the well-known zero-velocity update mechanism used in IMU- or MARG sensor-based gait analysis applications. While the visual fixation holds, the data fusion process will neglect gyroscope bias accumulation errors in the yaw estimate. In anticipation of only supporting the heading estimate and not influencing or changing the attitude estimation, a heading vector substitute mechanism is proposed. This method is implemented into an adapted version of the GDA filter stage, which enables a substitution of the magnetometer measurement with a heading vector based on IMU measurements only, effectively reducing the set of equations to the MARG sensor only case. Calculating a substitute vector to the magnetometer endows a new disturbance detection method based on vector scalar products, surpassing the prior developed quaternion difference method in its applicability. The approach does not rely on a world camera feed and therefore can be termed infrastructureless and self-contained. Nevertheless, the fusion process does experience limitations. The method delivers its greatest benefit during static or near-static states of motion. The more pupil fixations occur, the better the effect of overall heading drift reduction. Lesser fixation counts lead to a lower drift reduction. The second limitation stems from very slow rotational rates. Angular velocities of less than  $1^\circ/s$  could potentially derail the fixation detection plugin, falsely classifying the motion phase as static, if the eyes fixate a target. If we put this low-end threshold into perspective with the measurement range from a commercially available sensor at low angular rates (compare figure 4.5, FSM-9 angles at  $3^\circ s^{-1}$  rotational rate), one can observe that the chosen threshold is an adequate choice. In conclusion, the method is still applicable, as long as the angular velocities being measured exceed  $1^\circ/s$ . Using pupil position estimates alongside MARG sensors demonstrates to be a useful additional sensor measurement to estimate the head motion state. This is especially the case for low-cost high-noise consumer-grade MARG sensors, which get the maximum benefit from the proposed pipeline due to the high degree of inherent noise. During dynamic changes, the method does not invoke any improvements on the orientation estimation. Errors arising from coning motion (high vibrational motion) and angle random walk (ARW) will inevitably accumulate. In the case where the user fixates a target that is moving at the same angular velocity as the head is moving, the algorithm assumes a fixation and loop back the yaw estimate to the filter. This is, however, only the case for very slow angular rates lower than  $1^\circ/s$ , since the method employs a minimum angular rate threshold detection while a fixation is recognized. To account for this error accumulation further sources of reference are needed, enabling a backwards correction

---

of accumulation errors while magnetic measurements are not available.

Chapter 6 presents a method to integrate an eye tracker's scene camera by means of a state-of-the-art vSLAM framework into the developed orientation estimation pipeline. Using the eye tracker's scene camera alongside an appropriate computer vision approach solves the backwards correction problem. The approach incorporates a vSLAM system that feeds its orientation estimation to the MARG-sensor system and enables the goal of robust infrastructureless head orientation estimation under varying disturbances. Fusing orientation inside the MARG-sensor and not incorporating it into the SLAM system itself gives a modularization and customization in terms of using different SLAM or visual orientation estimation approaches, e.g. ORB-SLAM, LSD-SLAM, VO approaches or any other. Isolating the heading components of the quaternion, building a magnetometer substitute vector, and feeding it backwards to the IMU/MARG creates the possibility to reset the SLAM approach as a whole and use the current IMU/MARG orientation to initialize the pose estimation. Calculating a substitute to the magnetometer measurement from either the IMU substitute from pupil fixation detection or visual pose estimation from the ORB-SLAM framework, simplifies the data fusion process in such a way that the designed quaternion-based filter does not need to be tuned since the equations stay the same. The only thing changing is the sensor noise parameter influencing the covariance estimation during dynamic motion. Furthermore, it is possible to dynamically switch between the different heading sources to increase robustness based on the availability of the data streams, e.g. if the SLAM system does not recover the current pose in the localization mode or the magnetometer is heavily disturbed.

Since the implementation does currently not involve a position computation fusing inertial and visual sensors, the position might be subject to errors, if the SLAM does not find a good initial guess when it is reset or disturbed over a long period of time. This does interfere with the gaze point control mode since the position estimate could be erroneous. This however can be addressed by a reinitialization of the current pose using the initialization described in section 6.2.

The method presented relies on a dedicated hardware that enables the execution of the ORB-SLAM framework in real-time and takes care of the pupil service application. Due to the ongoing increase in computational performance, miniaturization of single board computers, and reduction in power consumption, it is only a matter of time and development efforts to enable a stand-alone and mobile application, e.g. using a Jetson Xavier alongside a LiPo battery. Combining MARG and a visual sensor system to extract robust pose estimations and using this pose information in the context for mobile eye tracking widens up the possible control signals for direct robot control. Using the above-presented methods leads to precise head or eye gaze-based Cartesian robot control, simplifying and delivering a more natural way of direct human-robot collaboration. The RGB-D camera used contributes to the overall weight by more than 2/3. If the eye- or head-gaze control channel is not required, the system could be used with a monocular camera (e.g. the built-in camera see figure 3.2b) effectively reducing the size and

weight. It would, however, require to change the ORB-SLAM method to the monocular case, which decreases overall accuracy. It should be further noted that the new implementation of ORB-SLAM (ORB-SLAM 3 [74]) features a full implementation for visual-inertial SLAM, including bias estimation and a multi-map SLAM system for increased robustness. The drawback that stems from the full implementation for viSLAM is the need for a very accurate calibration (transformation) between the visual and inertial reference frames and time synchronization. The system quickly diverges if these attributes cannot be guaranteed or are compromised. Furthermore, the VI-SLAM implementation does not use magnetometer measurements for heading corrections or absolute orientation initialization if the visual feedback gets corrupted, since cameras will always depend on reasonable lighting conditions.

However, the loosely coupled approach presented here does produce an estimation even with slight mismatches, and also enables magnetometer data to be used for absolute orientation initialization and heading correction if visual data is not reliable.

To conclude, the goal of this thesis is the development of a multimodal infrastructure-less sensor system for robust robot control. During the thesis a lightweight interface is developed combining a MARG sensor, a head-worn eye tracker and an RGB-D camera for head motion as well as head and eye gaze robot control. A robust orientation estimation filter to reduce the influence of short-time magnetic disturbances on MARG sensors is implemented. Furthermore, a novel method invoking visual fixations from eye tracking glasses is presented to reduce the inherent drift of a single sensing MARG sensor for long-term magnetic disturbances. A state-of-the-art vSLAM framework (ORB-SLAM) was integrated into the fusion process of the MARG sensor to achieve backwards correction for high robustness and enable 3D gaze point interaction. Finally, a proof of concept and functionality of the interface inside a human-robot use case concludes the work. The developed data fusion process and interface enable a more robust and precise human-robot collaboration by means of head motion control. The methods facilitate hands-free robot control by introducing head and eye gaze robot control by means of absolute 3D position control. Besides this, the user is also capable of controlling a robotic system in all rotational and translational DoFs.

It should be noted that it would nevertheless be the best possible option to only use MARG sensors and no additional hardware at all. This would allow for a small form factor of the sensor system (compare figure 3.2a) and therefore contribute to the overall user acceptance of the device. Unfortunately, even the market leader for MEMS based inertial motion capture (Xsens) cannot guarantee such an accuracy that would inherit no safety risk at all with a single MARG sensor.

## 9 Outlook

The developed head-worn interface and methods enable a robust head pose estimation in magnetically disturbed environments but nonetheless expose some potential improvements in various ways or fields.

**MARG sensors:** The ongoing research in the field of MEMS-based MARG sensors has led to an increase in the accuracy and reliability of inertial and magneto-inertial only orientation estimation. The advances in hardware specifications of the sensors directly influence the performance. This includes further advancements in temperature-based compensation. Since MARG sensors feature built-in temperature sensors, it could be beneficial to build a close-looped thermal regulation at the chip or sensor level. The sensor should be placed inside a closed confinement (housing) equipped with a metal meander to heat up the sensor system to a steady level. The measured chip temperature can be used to regulate the heating element. Since the volume that is to be heated is small, there should be no significant overshoot of the temperature regulation. A constant temperature at the chip level will ultimately result in a reduction of gyroscope bias (compare section 2.1.2) over a long period of time.

A further field of research regarding MARG sensor only orientation estimation is the use of machine learning approaches. A potential approach could include machine learning for fast magnetometer mapping. Therefore, magnetometer measurements could be sampled, alongside the heading vector, estimated using the IMU. The human user rotates the head in the maximal needed motion range (e.g.  $\pm 30^\circ$  in all axis). Both measurements could be used to train a small neural network that allows mapping of the raw magnetometer data onto the heading vector. The neural network would produce a valid magnetometer vector at least in the covered volume. This, however, is up to future research.

**MARG sensor and eye tracking data fusion:** Research on eye tracking methods and MARG sensor-based fusion will be focused in future work because of the potential it holds for AR applications. Novel sensor systems are currently developed to produce AR signals using laser scanning on a human's retina, without the need for any world cameras. These interfaces also enable tracking of pupil position data, which in turn could be used alongside MARG sensor data in combination with the developed approach to achieve robust orientation and eye tracking data with an AR GUI feedback in a small and self-contained package [93]. Besides this, further research should be conducted re-

garding an adaptive gyroscope noise parameter estimation based on the proposed visual fixation trigger for head motion detection. The gyroscope noise parameter estimation could be used to reduce the heading errors even further and without the visual fixation trigger being active. While a sufficient number of samples is gathered during visual fixations, a gyroscope bias guess can be estimated and passed to the MARG sensor. A second instance of the filter running in parallel could be used to compute an orientation that includes the estimated gyroscope bias and compare it to the first instance of the filter in real-time. Based on the deviation between both solutions, the estimated bias could be used or discarded from that point on, which in turn will lead to improved heading accuracy.

In addition, the research and development of data fusion between cameras and MARG sensors should naturally be pursued further. This particularly includes the reduction of the computational cost of SLAM and computer vision applications as well as the use of smaller monocular cameras to reduce the overall form factor of the system and further decrease weight.

It might furthermore be beneficial to investigate the potential data fusion of simultaneous eye and head motion for activation of the continuous head motion control. For example, one could activate the head motion control only while the eyes are fixating a point and a simultaneously measurement of head motion as well as the antagonistic eye motion is detected, e.g. using cross correlation to detect the event. This would a) counteract the Midas Touch problem and b) bypass the heading drift problem since the approach would not require a continuous head motion computation.

**Human-robot collaboration:** The presented interface enables head motion as well as head- and eye gaze control signals for direct robot teleoperation and collaboration tasks. These communication modalities enable a variety of new control approaches. For example, the eye gaze position estimation and head motion measurement could be used in direct combination to translate and rotate the robot's EEF at the same time without switching motion states, i.e., fixating a target object with the eyes and directly rotating the head in the desired orientation.

Furthermore, the Cartesian gaze point estimation could be used for semi-autonomous object grasping. The three-dimensional gaze position output could be used to set a Cartesian region of interest (ROI) for a robotic arm that is equipped with a wrist camera. The estimation of the ROI of the object enables the wrist camera on the robotic arm to accurately localize and grab the object based on machine learning or artificial intelligence, for example using the `robobrain.vision`<sup>6</sup> from robominds to localize and estimate the best pick point of an object in unsorted containers. This procedure could lead to a more natural way of human-robot collaboration by just gazing at an object of interest and thereby effectively reducing the number of control inputs or motion groups needed.

---

<sup>6</sup><https://www.robominds.de/robobrain-vision-2/>

---

It would also be beneficial to incorporate the eye gaze gesture recognition from [20] to increase the robustness of the switching mechanisms between the different control groups to contribute to the overall usability.

In addition to improvements of the sensor system and interface, data fusion with distributed heterogeneous sensor systems of the collaborating robot system could be explored and utilized. Among other things, a robot's wrist camera could be used to detect the orientation of the collaborating human's head via computer vision-based methods. This head orientation could be fused with the MARG sensor measured head orientation to, for example, uniquely assign a user within a group of people based on the redundant head movement measurements. This is however up to future research projects exploring these potential use cases and data fusion methods.



## Bibliography – authored and co-authored publications

- [1] L. Wöhle and M. Gebhard, “A robust quaternion based kalman filter using a gradient descent algorithm for orientation measurement,” in 2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), pp. 1–6, 2018.
- [2] S. Stalljann, L. Wöhle, J. Schäfer, and M. Gebhard, “Performance analysis of a head and eye motion-based control interface for assistive robots,” Sensors, vol. 20, no. 24, 2020.
- [3] L. Wöhle, S. Miller, J. Gerken, and M. Gebhard, “A robust interface for head motion based control of a robot arm using marg and visual sensors,” in 2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA), pp. 1–6, IEEE, 2018.
- [4] L. Wöhle and M. Gebhard, “Steadeye-head—improving marg-sensor based head orientation measurements through eye tracking data,” Sensors, vol. 20, no. 10, 2020.
- [5] L. Wöhle and M. Gebhard, “Towards robust robot control in cartesian space using an infrastructureless head- and eye-gaze interface,” Sensors, vol. 21, no. 5, 2021.
- [6] S. Schöllmann, P. Try, M. Gebhard, and L. Wöhle, “Sensors for assistive robotic drinking with physical contact,” in 2021 IEEE International Symposium on Medical Measurements and Applications (MeMeA), pp. 1–6, 2021.
- [7] P. Try, S. Schöllmann, L. Wöhle, and M. Gebhard, “Visual sensor fusion based autonomous robotic system for assistive drinking,” Sensors, 2021. Submitted to Special Issue Assistance Robotics and Sensors.



## Bibliography

- [8] D. Gorecky, M. Schmitt, M. Loskyll, and D. Zühlke, “Human-machine-interaction in the industry 4.0 era,” in 2014 12th IEEE International Conference on Industrial Informatics (INDIN), pp. 289–294, July 2014.
- [9] D. K. Kumar and S. P. Arjunan, Human-computer Interface Technologies for the Motor Impaired, vol. 12. CRC Press, 2015.
- [10] M. Kyrarini, A. Leu, D. Ristić-Durrant, A. Gräser, A. Jackowski, M. Gebhard, J. Nelles, C. Bröhl, C. Brandl, A. Mertens, et al., “Human-robot synergy for cooperative robots,” Facta Universitatis, Series: Automatic Control and Robotics, vol. 15, no. 3, pp. 187–204, 2016.
- [11] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, and P. Fraise, “Collaborative manufacturing with physical human–robot interaction,” Robotics and Computer-Integrated Manufacturing, vol. 40, pp. 1–13, 2016.
- [12] M. A. Goodrich and A. C. Schultz, Human-robot interaction: a survey. Now Publishers Inc, 2008.
- [13] S. Frennert, H. Aminoff, and B. Östlund, “Technological frames and care robots in eldercare,” International Journal of Social Robotics, pp. 1–15, 2020.
- [14] A. Graser, T. Heyer, L. Fotoohi, U. Lange, H. Kampe, B. Enjarini, S. Heyer, C. Fragkopoulos, and D. Ristic-Durrant, “A supportive friend at work: Robotic workplace assistance for the disabled,” IEEE Robotics Automation Magazine, vol. 20, no. 4, pp. 148–159, 2013.
- [15] T. L. Chen, M. Ciocarlie, S. Cousins, P. M. Grice, K. Hawkins, K. Hsiao, C. C. Kemp, C.-H. King, D. A. Lazewatsky, A. E. Leeper, H. Nguyen, A. Paepcke, C. Pantofaru, W. D. Smart, and L. Takayama, “Robots for humanity: using assistive robotics to empower people with disabilities,” IEEE Robotics Automation Magazine, vol. 20, no. 1, pp. 30–39, 2013.
- [16] A. Jackowski, M. Gebhard, and R. Thietje, “Head motion and head gesture-based robot control: A usability study,” IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 26, no. 1, pp. 161–170, 2018.

- 
- [17] N. Rudigkeit and M. Gebhard, “Amicus 2.0—system presentation and demonstration of adaptability to personal needs by the example of an individual with progressed multiple sclerosis,” *Sensors*, vol. 20, no. 4, 2020.
- [18] S. Alsharif, O. Kuzmicheva, and A. Gräser, “Gaze gesture-based human robot interface,” *Technische Unterstützungssysteme, die die Menschen wirklich wollen*, p. 339, 2016.
- [19] S. Li, X. Zhang, and J. D. Webb, “3-d-gaze-based robotic grasping through mimicking human visuomotor function for people with motion impairments,” *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 12, pp. 2824–2835, 2017.
- [20] S. Alsharif, *Gaze-Based Control of Robot Arm in Three-Dimensional Space*. PhD thesis, 2018.
- [21] N. Rudigkeit and M. Gebhard, “Amicus—a head motion-based interface for control of an assistive robot,” *Sensors*, vol. 19, no. 12, 2019.
- [22] J.-K. Shiau, C.-X. Huang, M.-Y. Chang, *et al.*, “Noise characteristics of mems gyro’s null drift and temperature compensation,” *Journal of Applied Science and Engineering*, vol. 15, no. 4, pp. 239–246, 2012.
- [23] A. Vydhyathan, G. Bellusci, H. Lunge, and P. Slycke, “The next generation xsens motion trackers for industrial applications,” *Xsens: Enschede, The Netherlands*, 2015.
- [24] Xsens, “Active heading stabilization (ahs).” [https://base.xsens.com/knowledgebase/s/article/Active-Heading-Stabilization-AHS-1605869706072?language=en\\_US](https://base.xsens.com/knowledgebase/s/article/Active-Heading-Stabilization-AHS-1605869706072?language=en_US). Accessed: 26 May 2021.
- [25] M. Caruso, A. M. Sabatini, D. Laidig, T. Seel, M. Knaflitz, U. Della Croce, and A. Cereatti, “Analysis of the accuracy of ten algorithms for orientation estimation using inertial and magnetic sensing under optimal conditions: One size does not fit all,” *Sensors*, vol. 21, no. 7, 2021.
- [26] N. Rudigkeit, *AMiCUS - Bewegungssensor-basiertes Human-Robot Interface zur intuitiven Echtzeit-Steuerung eines Roboterarmes mit Kopfbewegungen*. PhD thesis, 2017.
- [27] P. M. Tostado, W. W. Abbott, and A. A. Faisal, “3d gaze cursor: Continuous calibration and end-point grasp control of robotic actuators,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3295–3300, 2016.
- [28] L. Scalera, S. Seriani, A. Gasparetto, and P. Gallina, “A novel robotic system for painting with eyes,” in *Advances in Italian Mechanism Science* (V. Niola and A. Gasparetto, eds.), (Cham), pp. 191–199, Springer International Publishing, 2021.

- [29] S. Dziemian, W. W. Abbott, and A. A. Faisal, "Gaze-based teleprosthetic enables intuitive continuous control of complex robot arm use: Writing drawing," in 2016 6th IEEE International Conference on Biomedical Robotics and Biomechanics (BioRob), pp. 1277–1282, 2016.
- [30] A. Shafti, P. Orlov, and A. A. Faisal, "Gaze-based, context-aware robotic system for assisted reaching and grasping," in 2019 International Conference on Robotics and Automation (ICRA), pp. 863–869, 2019.
- [31] Microsoft, "Hololens (1st gen) hardware." <https://docs.microsoft.com/en-gb/hololens/hololens1-hardware>. Accessed: 1 December 2020.
- [32] Microsoft, "Hololens 2." <https://www.microsoft.com/en-us/hololens/hardware>. Accessed: 1 December 2020.
- [33] X. Robert-Lachaine, H. Mecheri, C. Larue, and A. Plamondon, "Effect of local magnetic field disturbances on inertial measurement units accuracy," Applied ergonomics, vol. 63, pp. 123–132, 2017.
- [34] C. Wong, Z.-Q. Zhang, B. Lo, and G.-Z. Yang, "Wearable sensing for solid biomechanics: A review," IEEE Sensors Journal, vol. 15, pp. 2747–2760, May 2015.
- [35] F. Wenk, Inertial Motion Capturing : Rigid Body Pose and Posture Estimation with Inertial Sensors. PhD thesis, 2017.
- [36] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in IEEE International Conference on Rehabilitation Robotics (ICORR), pp. 1–7, 2011.
- [37] J. Auysakul, H. Xu, and V. Pooneeth, "A hybrid motion estimation for video stabilization based on an imu sensor," Sensors, vol. 18, no. 8, 2018.
- [38] D. Kingston and R. Beard, "Real-time attitude and position estimation for small uavs using low-cost sensors," in AIAA 3rd unmanned unlimited technical conference, Workshop and exhibit, 2004.
- [39] G. Gupta, P. Barlow, and S. David, "Review of sensors and sensor integration for the control of a humanoid robot," in IEEE Instrumentation and Measurement Technology Conference (I2MTC), pp. 1–5, 2011.
- [40] W. Lang, Sensors and Measurement Systems. River Publishers Series in Electronic Materials and Devices, River Publishers, 2019.
- [41] J. Wendel, Integrierte Navigationssysteme: Sensordatenfusion, GPS und Inertiale Navigation. Walter de Gruyter, 2011.

- [42] C. Acar and A. Shkel, MEMS vibratory gyroscopes: structural approaches to improve robustness. Springer Science & Business Media, 2008.
- [43] T. InvenSense., “Icm-20948 —world’s lowest power 9-axis mems motiontracking device, icm-20948 datasheet..” <https://invensense.tdk.com/wp-content/uploads/2016/06/DS-000189-ICM-20948-v1.3.pdf>. Accessed: 15 April 2020.
- [44] M. Kok, J. D. Hol, T. B. Schön, F. Gustafsson, and H. Luinge, “Calibration of a magnetometer in combination with inertial sensors,” in 2012 15th International Conference on Information Fusion, pp. 787–793, 2012.
- [45] O. J. Woodman, “An introduction to inertial navigation,” tech. rep., University of Cambridge, Computer Laboratory, 2007.
- [46] “Ieee standard specification format guide and test procedure for single-axis laser gyros,” IEEE Std 647-2006 (Revision of IEEE Std 647-1995), pp. 1–96, Sep. 2006.
- [47] V. Renaudin, M. H. Afzal, and G. Lachapelle, “Complete triaxis magnetometer calibration in the magnetic domain,” Journal of sensors, vol. 2010, 2010.
- [48] T. Pylvänäinen, “Automatic and adaptive calibration of 3d field sensors,” Applied Mathematical Modelling, vol. 32, no. 4, pp. 575–587, 2008.
- [49] D. Turner, I. Anderson, J. Mason, and M. Cox, “An algorithm for fitting an ellipsoid to data,” National Physical Laboratory, UK, 1999.
- [50] R. Valenti, N. Sebe, and T. Gevers, “What are you looking at?,” International journal of computer vision, vol. 98, no. 3, pp. 324–334, 2012.
- [51] M. Kassner, W. Patera, and A. Bulling, “Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction,” in Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication, pp. 1151–1160, 2014.
- [52] A. Sprenger, B. Neppert, S. Köster, S. Gais, D. Kömpf, C. Helmchen, and H. Kimmig, “Long-term eye movement recordings with a scleral search coil-eyelid protection device allows new applications,” Journal of neuroscience methods, vol. 170, no. 2, pp. 305–309, 2008.
- [53] L. Larsson, A. Schwaller, K. Holmqvist, M. Nyström, and M. Stridh, “Compensation of head movements in mobile eye-tracking data using an inertial measurement unit,” in Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, pp. 1161–1167, 2014.
- [54] P. L. GmbH, “Pupil core. open source eye tracking platform home page.” <https://pupil-labs.com/products/core/>. Accessed: 12 April 2020.

- [55] P. L. GmbH, “Pupil invisible. the world’s first deep learning powered eye tracking glasses..” <https://pupil-labs.com/products/invisible/>. Accessed: 18 May 2021.
- [56] M. Tonsen, C. K. Baumann, and K. Dierkes, “A high-level description and performance evaluation of pupil invisible,” 2020.
- [57] K. Dierkes, M. Kassner, and A. Bulling, “A fast approach to refraction-aware eye-model fitting and gaze prediction,” in Proceedings of the 11th ACM Symposium on Eye Tracking Research and Applications, ETRA ’19, (New York, NY, USA), Association for Computing Machinery, 2019.
- [58] C. Mestre, J. Gautier, and J. Pujol, “Robust eye tracking based on multiple corneal reflections for clinical applications,” Journal of Biomedical Optics, vol. 23, no. 3, pp. 1 – 9, 2018.
- [59] R. O. Maimon-Mor, J. Fernandez-Quesada, G. A. Zito, C. Konnaris, S. Dziemian, and A. A. Faisal, “Towards free 3d end-point control for robotic-assisted human reaching using binocular eye tracking,” in 2017 International Conference on Rehabilitation Robotics (ICORR), pp. 1049–1054, 2017.
- [60] A. V. T. GmbH, “Ccd or cmos: can cmos sensors replace ccds in all cases?.” [https://www.stemmer-imaging.com/media/uploads/cameras/avt/12/120483-Allied\\_Vision\\_White\\_Paper\\_CCD\\_vs\\_CMOS.pdf](https://www.stemmer-imaging.com/media/uploads/cameras/avt/12/120483-Allied_Vision_White_Paper_CCD_vs_CMOS.pdf). Accessed: 08 June 2021.
- [61] L. Wang, Z. Zhang, and P. Sun, “Quaternion-based kalman filter for ahrs using an adaptive-step gradient descent algorithm,” International Journal of Advanced Robotic Systems, vol. 12, no. 9, p. 131, 2015.
- [62] H. A. Hashim, “Special orthogonal group  $so(3)$ , euler angles, angle-axis, rodriguez vector and unit-quaternion: Overview, mapping and challenges,” 2021.
- [63] X. Yun and E. R. Bachmann, “Design, implementation, and experimental results of a quaternion-based kalman filter for human body motion tracking,” IEEE transactions on Robotics, vol. 22, no. 6, pp. 1216–1227, 2006.
- [64] R. Mahony, T. Hamel, and J.-M. Pflimlin, “Nonlinear complementary filters on the special orthogonal group,” IEEE Transactions on Automatic Control, vol. 53, pp. 1203–1218, June 2008.
- [65] R. G. Valenti, I. Dryanovski, and J. Xiao, “Keeping a good attitude: A quaternion-based orientation filter for imus and margs,” Sensors, vol. 15, no. 8, pp. 19302–19330, 2015.

- 
- [66] D. Scaramuzza and F. Fraundorfer, “Visual odometry [tutorial],” *IEEE Robotics Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.
- [67] P. Moulon, P. Monasse, R. Marlet, and Others, “Openmvg.” <https://github.com/openMVG/openMVG>.
- [68] F. Hu, J. Cheng, Y. Bao, and Y. He, “Accuracy enhancement for the front-end tracking algorithm of rgb-d slam,” *Intelligent Service Robotics*, vol. 13, no. 2, pp. 207–218, 2020.
- [69] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, “An overview to visual odometry and visual slam: Applications to mobile robotics,” *Intelligent Industrial Systems*, vol. 1, no. 4, pp. 289–311, 2015.
- [70] H. Strasdat, J. Montiel, and A. Davison, “Visual slam: Why filter?,” *Image Vis. Comput.*, vol. 30, pp. 65–77, 2012.
- [71] F. Fraundorfer and D. Scaramuzza, “Visual odometry : Part ii: Matching, robustness, optimization, and applications,” *IEEE Robotics Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.
- [72] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part i,” *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.
- [73] T. Bailey and H. Durrant-Whyte, “Simultaneous localization and mapping (slam): part ii,” *IEEE Robotics Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.
- [74] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, “Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam,” 2021.
- [75] R. Mur-Artal and J. D. Tardós, “Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.
- [76] I. Corporation, “Intel realsense d400 series product family, datasheet.” <https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/Intel-RealSense-D400-Series-Datasheet.pdf>. Accessed: 12 April 2020.
- [77] D. Gebre-Egziabher, G. H. Elkaim, J. David Powell, and B. W. Parkinson, “Calibration of strapdown magnetometers in magnetic field domain,” *Journal of Aerospace Engineering*, vol. 19, no. 2, pp. 87–102, 2006.
- [78] P. L. GmbH, “Pupil labs. about pupil labs home page.” <https://pupil-labs.com/about/>. Accessed: 22 June 2021.

- [79] R. G. Valenti, I. Dryanovski, and J. Xiao, "A linear kalman filter for marg orientation estimation using the algebraic quaternion algorithm," IEEE Transactions on Instrumentation and Measurement, vol. 65, no. 2, pp. 467–481, 2016.
- [80] S. Wilson, H. Eberle, Y. Hayashi, S. O. Madgwick, A. McGregor, X. Jing, and R. Vaidyanathan, "Formulation of a new gradient descent marg orientation algorithm: Case study on robot teleoperation," Mechanical Systems and Signal Processing, vol. 130, pp. 183–200, 2019.
- [81] S. Yean, B. S. Lee, C. K. Yeo, C. H. Vun, and H. L. Oh, "Smartphone orientation estimation algorithm combining kalman filter with gradient descent," IEEE Journal of Biomedical and Health Informatics, vol. 22, pp. 1421–1433, Sep. 2018.
- [82] D. Gebre-Egziabher, G. Elkaim, J. D. Powell, and B. Parkinson, "A non-linear, two-step estimation algorithm for calibrating solid-state strapdown magnetometers," in 8th International St. Petersburg Conference on Navigation Systems (IEEE/AIAA), 2001.
- [83] Z. Wu, Z. Sun, W. Zhang, and Q. Chen, "Attitude and gyro bias estimation by the rotation of an inertial measurement unit," Measurement Science and Technology, vol. 26, no. 12, p. 125102, 2015.
- [84] Y. Wu, D. Zou, P. Liu, and W. Yu, "Dynamic magnetometer calibration and alignment to inertial sensors by kalman filtering," IEEE Transactions on Control Systems Technology, vol. 26, no. 2, pp. 716–723, 2018.
- [85] M. O. Aqel, M. H. Marhaban, M. I. Saripan, and N. B. Ismail, "Review of visual odometry: types, approaches, challenges, and applications," SpringerPlus, vol. 5, no. 1, pp. 1–26, 2016.
- [86] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in Proceedings of the 2000 symposium on Eye tracking research & applications, pp. 71–78, 2000.
- [87] A. T. Duchowski and S. Jörg, Eye Animation, pp. 1–19. Cham: Springer International Publishing, 2020.
- [88] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," Matrix, vol. 58, no. 15-16, pp. 1–35, 2006.
- [89] R. T. E. Ltd, "Freertos. real-time operating system for microcontrollers.." [https://www.freertos.org/Documentation/RTOS\\_book.html](https://www.freertos.org/Documentation/RTOS_book.html). Accessed: 12 April 2020.
- [90] STMicroelectronics, "Parameters and calibration of a low-g 3-axis accelerometer, application note 4508." [https://www.st.com/resource/en/application\\_note/dm00119044-parameters-and-calibration-of-a-lowg-3axis-accelerometer](https://www.st.com/resource/en/application_note/dm00119044-parameters-and-calibration-of-a-lowg-3axis-accelerometer)

protect\discretionary{\char\hyphenchar\font}{\char\hyphenchar\font}{\char\hyphenchar\font}.pdf. Accessed: 12 April 2020.

- [91] O. dev team., “Camera calibration and 3d reconstruction. pinhole camera model..” [https://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html](https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html). Accessed: 2 November 2020.
- [92] J. Wang and E. Olson, “Apriltag 2: Efficient and robust fiducial detection,” in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4193–4198, 2016.
- [93] J. Meyer, T. Schlebusch, W. Fuhl, and E. Kasneci, “A novel camera-free eye tracking sensor for augmented reality based on laser scanning,” IEEE Sensors Journal, vol. 20, pp. 15204–15212, Dec 2020.

## List of Figures

2.1	Schematic of MEMS based accelerometers. . . . .	13
2.2	Schematic of MEMS based gyroscope . . . . .	15
2.3	Schematic of the Hall effect. . . . .	16
2.4	Bias fluctuations of single-axis accelerometer data. . . . .	18
2.5	Bias fluctuations of single axis gyroscope data I. . . . .	22
2.6	Bias fluctuations of single axis gyroscope data II. . . . .	23
2.7	Allan deviation plot of a tri-axis gyroscope of ICM20948 at $50^{\circ}C$ . . . . .	25
2.8	Numerical integration errors from angular rate measurements. . . . .	26
2.9	Measurement values for a calibrated magnetometer in a constant mag- netic field . . . . .	27
2.10	Hard iron effect from fixed permanent magnetic source. . . . .	28
2.11	Hard iron effect from non-fixed permanent magnetic source. . . . .	30
2.12	Soft iron effect from fixed ferromagnetic source. . . . .	31
2.13	Soft iron effect from non-fixed ferromagnetic source. . . . .	32
2.14	Magnetometer measurements and the hard as well as soft iron corrections. . . . .	34
2.15	Simplified block diagram of MARG sensor based strapdown algorithm. . . . .	36
2.16	Pupil Labs eye detection pipeline. . . . .	40
2.17	Coordinate frames, I . . . . .	43
2.18	Coordinate frames, II . . . . .	44
2.19	Euler Angle representation. . . . .	47
2.20	Coordinate system transformation. . . . .	48
2.21	Unit-quaternion representation in 3D space. . . . .	50
2.22	Vector transformation from quaternion multiplication. . . . .	55
2.23	Illustration of incremental visual odometry estimation. . . . .	57
2.24	General VO and V-SLAM illustrations. . . . .	59
3.1	Block diagram of the overall system setup and general concept. . . . .	64
3.2	Modular sensor system design. . . . .	68
3.3	head and eye gaze depiction . . . . .	69
3.4	The WiOT system. . . . .	70
3.5	Block diagram of embedded software framework of the WiOT system. . . . .	72
3.6	Pupil Core headset configurations. . . . .	77
4.1	Simplified block diagram of the MARG filter fusion stage. . . . .	79
4.2	Depiction of the north direction vector calculation. . . . .	83

---

4.3	Block diagram of the developed core filter framework. . . . .	89
4.4	Typical results for ground truth and estimated angle at different angular rates. . . . .	92
4.5	Typical results magnetic disturbance rejection. . . . .	93
4.6	Results for magnetic disturbance rejection. . . . .	93
4.7	Error graph for magnetic disturbance rejection. . . . .	94
5.1	Simplified block diagram of the SteadEYE-head data fusion stage. . . . .	97
5.2	SteadEYE-head working principle. . . . .	98
5.3	Head- and eye-motion dependencies II. . . . .	99
5.4	Head- and eye-motion dependencies I. . . . .	100
5.5	North direction vector substitute from quaternion multiplication. . . . .	107
5.6	Block diagram of the eye tracking-supported MARG sensor data fusion approach. . . . .	108
5.7	ROS nodes for MARG and eye tracker synchronization. . . . .	109
5.8	Experimental setup for SteadEYE-head measurements. . . . .	110
5.9	Vector substitute comparisons. . . . .	112
5.10	Magnetic disturbance influence and detection. . . . .	114
5.11	Sequence for yaw angle estimation results with SteadEYE-Head fusion. . .	115
5.12	Yaw angle error graphs for 8 different trials. . . . .	116
6.1	Simplified block diagram of the complete data fusion framework. . . . .	121
6.2	Different camera image streams. . . . .	124
6.3	Block diagram of the complete data fusion process. . . . .	129
6.4	Block diagram of the 3D gaze point estimation pipeline. . . . .	130
6.5	3D gaze point reconstruction. . . . .	132
6.6	Experimental Setup for gaze point and pose accuracy estimation. . . . .	134
6.7	Yaw angle results of the developed data fusion approach. . . . .	136
6.8	Sequence of yaw angle estimations during complete loss of visual data. . .	137
6.9	Results for head trajectory estimations for one trial. . . . .	139
6.10	Measurement results for head gaze accuracy. . . . .	141
6.11	Measurement results for eye gaze accuracy. . . . .	143
7.1	Full wedge adjuster assembly process (Keilsteller) . . . . .	146
7.2	Dual-arm robotic workspace. . . . .	147
7.3	Robot motion control groups. . . . .	149
7.4	6-DoF robot control block diagram . . . . .	150
7.5	3D gaze point robot control reprojction. . . . .	152
7.6	Head gaze results for robot control application. . . . .	153
7.7	Robot application. . . . .	154
7.8	Trajectory and orientation depiction of single part isolation and placement I.	155

7.9 Trajectory and orientation depiction of single part isolation and placement  
II. . . . . 155



## List of Tables

2.1	Noise parameter for ICM-20948 gyroscope. . . . .	26
2.2	Comparison of Kalman vs. Complementary Filter. . . . .	56
2.3	Comparison of VO vs. V-SLAM. . . . .	60
3.1	Technical specifications of the WiOT system. . . . .	75
3.2	Technical Specifications of pupil core eye tracker. . . . .	77
4.1	RMS values for KFGDA and commercial data fusion. . . . .	91
5.1	Absolute heading error values with and without eye tracking support. . . . .	117
6.1	Mean of RMSE values for inertial and the proposed visual-inertial orientation estimation. . . . .	135
6.2	Mean of RMSE values for visual position estimation. . . . .	138
6.3	Mean of RMSE values for gaze position estimations. . . . .	140
7.1	Motion parameters for the UR5 teleoperation. . . . .	152

# Danksagungen

Ein besonderer Dank gilt Frau Professor Dr. Marion Gebhard, die es mir ermöglicht hat, diese Promotion innerhalb der Arbeitsgruppe SAM (Sensorik und Aktorik in der Medizintechnik) der Westfälischen Hochschule Gelsenkirchen durchzuführen. Über die vielen Freiräume, den intensiven Austausch und die großartige Betreuung während der gesamten Forschungszeit bin ich überaus dankbar.

Darüber hinaus möchte ich mich bei meinem Doktorvater Herrn Professor Dr. Axel Gräser für die wissenschaftliche Betreuung sowie Begutachtung dieser Dissertation bedanken, wodurch ich überhaupt erst die Möglichkeit zu dieser Promotion bekam.

Weiterhin gilt mein Dank Herrn Prof. Dr. Walter Lang, der sich bereit erklärt hat diese Dissertation als Zweitgutachter zu prüfen und die Forschungsarbeiten durch konstruktives Feedback verbessert hat.

Des Weiteren möchte ich mich bei den Mitarbeitern der Arbeitsgruppe SAM der Westfälischen Hochschule Gelsenkirchen bedanken, die mir über die gesamte Dauer der Promotion mit vielen Ratschlägen und Hilfestellungen zur Seite standen. Aus Arbeitskollegen sind in dieser Zeit Freunde geworden.

Mein besonderer Dank gilt hierbei Alina Zickmann, die mir den Rücken freigehalten hat, sodass ich mich vollständig auf die Inhalte der Promotion und des Projektes MiA konzentrieren konnte, ohne mich von den Verwaltungsaufgaben ablenken zu lassen.

Auch bei meinem Büronachbarn Joscha Siewert bedanke ich mich – für den konstruktiven Austausch und die Mitarbeit an den unterschiedlichen ROS-Software-Komponenten, insbesondere der Anbindung der Pupil Labs Eyetracking Brille in einen C++-basierten ROS node.

Bei Andreas Künstler möchte ich mich ganz herzlichen bedanken für die gute Zusammenarbeit als maßgeblicher Testproband und die hohe Frustrationstoleranz. Es hat mir immer viel Spaß gemacht, die Algorithmen live zu testen und das Feedback von Andreas in die nächste Iteration einfließen zu lassen.

Ein großer Dank gilt Steffen Schöllmann und Pieter Try, mit denen ich mich als Co-Author und Arbeitskollege austauschen und spannende Forschungsarbeiten betreiben durfte.

Weiterhin möchte ich mich bei der Firma Büngern Technik für die Zusammenarbeit und den Austausch während der Projektlaufzeit bedanken, obwohl dieser Austausch durch die Corona-Pandemie bedauerlicherweise nicht mehr persönlich stattfinden konnte.

Ganz besonders möchte ich mich bei meiner Familie bedanken, für die kontinuierliche Unterstützung während der gesamten Dauer der Arbeit. Das gilt besonders für meine

Frau Lisa Heiß-Wöhle, die mich durch die Höhen und Tiefen der Promotion begleitet und mir immer den Rücken gestärkt hat.

Zu guter Letzt möchte ich mich bei all denen bedanken, die zur Fertigstellung dieser Arbeit beigetragen haben, in welcher Form auch immer, und die ich vergessen bzw. namentlich hier nicht genannt habe.