State Estimation for Legged Robots - Kinematics, Inertial Sensing, and Computer Vision

Author(s):
Bloesch, Michael

Publication Date:
2017

Permanent Link:

https://doi.org/10.3929/ethz-a-010875968 →

DISS. ETH NO. 24130

State Estimation for Legged Robots – Kinematics, Inertial Sensing, and Computer Vision

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

Michael Andre Bloesch

MSc. ETH ME

born on August 4, 1987
citizen of Moerigen BE

accepted on the recommendation of

Prof. Dr. Roland Siegwart, Examiner
Prof. Dr. Andrew Davison, Co-examiner
Prof. Dr. Marco Hutter, Co-examiner

2017

Autonomous Systems Lab
Department of Mechanical and Process Engineering
ETH Zurich
Switzerland

Abstract

The primary aim of this thesis is to endow legged robots with a reliable sense of ego-motion. Just like humans or other legged beings, many legged robots require estimates of their posture and velocity in order to keep balance and move through the environment. The estimates need to exhibit both sufficiently high bandwidth and accuracy in order to allow for a controlled execution of these tasks. Furthermore, due to this dependency, failures of the state estimation may quickly lead to damaging of the robot or its surroundings, which emphasizes the importance of the reliability of the employed estimation algorithms. The associated research question may be formulated as finding an appropriate combination of sensor modalities and state estimation algorithms such that the ego-motion can reliably and accurately be estimated with financially and computationally reasonable costs. Furthermore, the state estimation should not limit the capabilities of the robot and thus the use of restrictive assumptions such as a horizontal terrain, specific gait patterns, or the availability of external sensing is undesirable.

In a first step the focus will be set on proprioceptive sensing in order to keep the data processing simple and thus keep time delays small and avoid additional error sources. In contrast to other types of robots, legged robots interact with their environment through intermittent foot-ground contacts. Assuming stationary ground contacts and measurable forward kinematics, this interaction can provide the legged robot with a very valuable source of information. Since most modern robotic platforms are also equipped with inertial sensing devices, the combination of inertial and kinematic data becomes an apparent approach for solving the state estimation problem for legged robots. However, careful modeling and sensor fusion design are prerequisites for achieving good performance. Concurrently, further challenges such as online calibration of Inertial Measurement Unit (IMU) biases, handling of slipping feet, or mitigating numerical inconsistencies have to be taken care of before truly reliable ego-motion estimation can be attained. In order to tackle these challenges we propose to co-estimate the foothold locations within a Kalman filter framework. We thereby also circumvent the use of restrictive assumptions such as a flat environment or a fixed gait pattern.

In a second part we investigate the use of exteroceptive sensing in order to reduce drift that occurs when employing proprioception only. We focus on the use of cameras due to the rich information they provide while exhibiting low weight and low power consumption. In order to achieve high robustness, it is important to incorporate inertial data during visual processing for reducing vision-only related failure modes, such as caused by fast motion or missing texture. Furthermore, the handling of fast motion during which the scene may pass rapidly trough the field of view, is improved if visual information extraction occurs from a feature's second observation onwards. Ideally, the handling of visual features is kept simple such that no cumbersome initialization

routine is required, which again improves robustness since the system can easily be reset if failures occur. Targeting these design concepts a first framework tightly combines optical flow measurements with inertial data. To this end an optical flow based residual is derived which relies on the co-estimation of the mean scene depth. The residual is then integrated in the update step of an IMU-driven Kalman filter.

Based on a similar IMU-driven Kalman filter approach we also investigate the possibility to tightly integrate the photometric information itself instead of relying on a visual pre-processing. The idea is to use the raw pixel intensity measurements directly in the Kalman filter update step by associating every landmark with a multilevel image patch. In subsequent camera frames, a photometric residual is derived by projecting the previously extracted patches into the images and computing a pixel-wise intensity error. If used within an Iterated Extended Kalman Filter (IEKF), this process directly takes care of the landmark tracking and no additional data association method is required. Furthermore, since this inherent landmark tracking relies on the use of inertial and visual information simultaneously it allows the inclusion of non-corner visual features such as line segments. The overall filter framework is formulated in a fully robot-centric way where landmark locations are partitioned into bearing vectors and distance parameters. This allows an undelayed and stochastically accurate initialization of new landmarks leading to a truly power-up-and-go state estimation framework.

Strong emphasis is set on the consistency and cleanliness of the developed methods. To this end differential geometric concepts are employed for the representation and handling of non-vector space quantities such as three-dimensional (3D) orientations and bearing vectors. The application of these concepts allows a minimal representation of differences and derivatives and thereby decreases the computational costs while leading to simple and singularity-free state estimation models. Furthermore, the corresponding "minimal" Jacobians can be used for performing a nonlinear observability analysis in order to identify the observable sub-space.

All state estimation algorithms are evaluated on real datasets. In many cases they have also been implemented on real robots and employed for feedback control. For instance, the proposed kinematic and inertial sensor fusion approach has become an inherent part of the software framework running on the quadrupedal robots StarlETH and ANYmal (see Figure 2.1). Likewise, the proposed visual inertial odometry has been applied in various Unmanned Aerial Vehicle (UAV)-related projects and is available as open-source software.

Zusammenfassung

Das primäre Ziel dieser Arbeit ist es, Laufroboter mit einem zuverlässigen Bewegungssinn zu versehen. Ähnlich wie Menschen und andere gehende Lebewesen, vertrauen viele Laufroboter auf eine Schätzung ihrer Haltung und Geschwindigkeit, um das Gleichgewicht zu halten und sich durch die Umwelt zu bewegen. Diese Schätzung muss eine ausreichend hohe Bandbreite und Genauigkeit aufweisen um eine kontrollierte Ausführung dieser Aufgaben zu gewährleisten. Gleichzeitig, stellt diese Abhängigkeit eine hohe Anforderung an die Zuverlässigkeit des Schätzalgorithmus. Ein Versagen der Zustandsschätzung kann dabei schnell zu einer Beschädigung des Roboters und seiner Umgebung führen. Die vorliegende Fragestellung richtet sich dementsprechend auf das Erforschen geeigneter Sensormodalitäten und entsprechender Schätzalgorithmen, so dass die Bewegung des Robotersystems zuverlässig und effizient geschätzt werden kann. Zusätzlich sollten die Fähigkeiten des Roboters nicht eingeschränkt werden und dementsprechend muss der Gebrauch von restriktiven Annahmen bezüglich Roboterumgebung oder Gangarten vermieden werden.

In einem ersten Teil fokussiert sich die Arbeit auf das Verarbeiten von propriozeptiven Messdaten. Dies erleichtert die Datenverarbeitung und ermöglicht dadurch, Zeitverzögerungen gering zu halten und zusätzliche Fehlerquellen zu vermeiden. Im Gegensatz zu anderen Roboterarten interagieren Laufroboter mittels intermittierenden Bodenkontakten mit ihrer Umgebung. In der Annahme dass Bodenkontakte stationär bleiben, stellt diese Interaktion eine sehr wertvolle Informationsquelle dar. Dies erfordert natürlich, dass die Vorwärtskinematik der Beine mittels kinematischen Sensoren erfasst werden kann. Da die meisten modernen Roboter auch mit sogenannten IMUs (Trägheitsnavigationssysteme) ausgestattet sind, ergibt sich die Kombination von Trägheits- mit kinematischen Messungen als effizienter Ansatz für die Zustandsschätzung von Laufrobotern. Allerdings bedingt dies eine sorgfältige Modellierung und den Einsatz adäquater Schätzalgorithmen. Zusätzlich müssen weitere Schwierigkeiten wie die online Kalibrierung von systematischen IMU Messabweichungen, die Handhabung von rutschenden Füssen, oder das Vorbeugen von Inkonsistenzen angegangen werden, bevor eine zuverlässige Bewegungsschätzung erreicht werden kann. Um diese Schwierigkeiten anzugehen, schlagen wir vor, die Schätzung der Fusspunkte im Schätzalgorithmus mit einzubeziehen. Damit vermeiden wir auch den Einsatz restriktiver Annahmen wie etwa einer flachen Umgebung oder einer vorgegeben Gangart.

Im zweiten Teil wird die Verwendung von exterozeptiven Sensoren zur Driftreduktion untersucht. Dabei wird der Fokus auf die Verwendung von Kameras gesetzt, da diese leichten und energiesparenden Sensoren einen sehr reichen Informationsgehalt aufweisen und über die Wiederbeobachtungen von Orientierungspunkten eine Driftreduktion ermöglichen. Um eine hohe Zuverlässigkeit zu erreichen, ist es äusserst wichtig, während der Verarbeitung visueller Daten die Messungen einer vorhandenen IMU mitzuberücksichtigen. Dies vermindert die Anfälligkeit auf visuell degenerierte

Situationen, wie sie beispielsweise bei schnellen Bewegungen oder fehlender Umgebungstextur anzutreffen sind. Bei schnellen Bewegungen ist es wichtig, dass die Information bereits aus wenigen Beobachtungen einzelner Bildpunkte extrahiert werden kann, da die Umgebungen sehr schnell an der Kamera vorbeizieht. Eine einfache und unverzögerte Handhabung der visuellen Orientierungspunkte ist darum unabdingbar. Zusätzlich kann dies auch die Initialisierungsroutine vereinfachen und dadurch ein Zurücksetzen des Systems bei Fehlern ermöglichen. Um diese Konzepte zu verfolgen, werden in einem ersten Ansatz IMU daten mit sogenannten optischen Fluss Messungen kombiniert. Hierzu wird der optische Fluss in ein Residuum umgewandelt, indem eine Annahme über die Szenentiefe getroffen wird. Dieses Residuum wird dann in den Aktualisierungsschritt eines Kalman Filter integriert, dessen Prädiktionsschritt auf IMU Daten basiert.

Gestützt auf eine ähnliche IMU-Prädiktion, wurden auch Kalman Filter basierte Schätzer untersucht, mit denen photometrische Information direkt integriert wird. Anstatt auf eine visuelle Vorverarbeitung zu vertrauen, besteht die Idee darin, die Bildintensitätsmessungen direkt im Aktualisierungsschritt des Kalman Filters zu verarbeiten. Dies wird erreicht indem jeder Orientierungspunkt mit einer kleinen quadratischen Bildregion assoziiert wird. Diese Bildregion wird in nachfolgenden Kameraaufnahmen mit den Bildintensitäten verglichen an der Stelle, wo der Orientierungspunkt zu erwarten ist, um einen pixelweisen Intensitätsfehler zu berechnen. Innerhalb eines iterierten erweiterten Kalman filters (IEKF) kann der Intensitätsfehler während des Aktualisierungsschritt direkt als Innovation benutzt werden. Das Interessante dabei ist, dass dadurch die Orientierungspunktverfolgung zu einem inhärenten Bestandteil des Kalman filters wird und dass keine zusätzliche Datenassoziation benötigt wird. Des weiteren werden die IMU-daten somit auch während dieser inhärenten visuellen Datenassoziation berücksichtigt was wiederum die Zuverlässigkeit erhöht und die Verfolgung von nicht Eckpunkten ermöglicht (zum Beispiel Liniensegmente). Eine vollständige roboterzentrische Formulierung wird benutzt, welche Orientierungspunkte in Richtungsvektoren und Distanzparameter zerlegt. Dies ermöglicht eine unverzögerte und stochastisch genaue Initialisierung von neuen Orientierungspunkten, so dass ein echtes "Power-Up-and-Go" system erreicht werden kann.

In dieser Arbeit wird stark auf Richtigkeit und Einfachheit der entwickelten Methoden geachtet. Dazu werden differentiell geometrische Konzepte für die Darstellung und Handhabung von nicht-Vektorraumgrössen wie 3D Orientierungen und Richtungsvektoren verwendet. Die Anwendung dieser Konzepte erlaubt eine minimale Darstellung von Differenzen und Ableitungen, verringert die Rechenkosten, und führt zu einfachen und singularitätsfreien Modellen. Weiterhin können die entsprechenden "minimalen" Jacobians in einer nichtlinearen Beobachtbarkeitsanalyse verwendet werden, um den beobachtbaren Unterraum zu identifizieren.

Die entwickelten Schätzalgorithmen werden auf realen Datensätzen ausgewertet. Zusätzlich werden einige der vorgeschlagenen Ansätze auf realen Robotern implementiert und dabei täglich gebraucht. Zum Beispiel ist der auf kinematischen und inertialen Daten basierende Ansatz zu einem inhärenten Teil der Softwareumgebung geworden, die auf den vierbeinigen Robotern StarlETH und ANYmal läuft (siehe Figure 2.1). Ebenso wurde die vorgeschlagene visuelle-inertiale Odometrie in verschiedenen Flugroboter bezogenen Projekten angewendet und ist als open-source Software verfügbar.

Acknowledgements

First and foremost, I would like to thank my advisor Prof. Roland Siegwart for his continuous motivation, for his open-mindedness and support of fancy ideas, and especially for the opportunity to pursue my doctoral thesis at the autonomous systems lab. It is an extraordinary research environment which allows its members to work in an extremely affectionate and exciting atmosphere while pursuing top level research.

Furthermore, I would like to extend my thanks to all other research groups I could share time with, what allowed me to gain very valuable experience. This of course includes the labs of both co-supervisors, Prof. Marco Hutter and Prof. Andrew Davison, where I spent a significant amount of time. But I would also like to thank Prof. Darwin Caldwell, Claudio Semini, Stephane Bazeille, Michele Focchi, Prof. Jonas Buchli, Michael Neunert, Prof. Davide Scaramuzza, Prof. Stefan Schaal, Ludovic Righetti and Nicholas Rotella for the fruitful collaborations and great times during extended academic exchanges.

This thesis would not have been possible without the help of fellow lab members and friends. Tending towards remaining too theoretical, I must express my deepest gratitude to all colleagues helping me with the more practical tasks of my doctoral studies. This particularly includes Christian Gehring, Peter Fankhauser, Dario Bellicoso and Remo Diethelm for their continuous help with experiments on both quadrupedal robots StarlETH and ANYmal. Furthermore I would like to thank Jörn Rehder and Janosh Nikolic for their support in dealing with the VI-sensor and Michael Burri for the help during the recording of numerous flight datasets. However, my sincerest gratitude goes also to Sammy Omari, Hannes Sommer, Stefan Leutenegger, and Gabriel Nützi for the many inspiring discussions, the mathematical support and for the creative and lively brain-storming sessions.

There are innumerable further lab mates I am indebted to and this goes beyond mere work-related aspects and includes everybody who contributed to a positive and joyful lab atmosphere. Especially, I would like to thank the "legged" team for the inspiring coffee breaks, for the many fun discussions, and the exciting moments spent together at international conferences. I also wish to extend my gratitude to the entire autonomous systems lab team for the many enjoyable lab events, including barbecues, aperos, ski-weekends and other lab-excursions.

Finally, and most important, I want to thank my family and friends as well as my partner Ingrid. I strongly believe in a good social and familial environment being a key factor for a fulfilling life. In this sense I thank you all full heartedly.

Zurich, February 2017 *Michael Bloesch*

Contents

Preface

Relevant work published during the course of the Author's doctoral studies make up the core of this *cumulative* thesis. This is framed by extended introduction and conclusion sections, which provide an overview of the work and illustrate how the different publications are part of a coherent research framework.

Chapter 1 begins with an introduction into the matter and states the motivation of the research. It also explains the division of the work into two parts. Subsequently, Chapter 2 discusses the current state-of-the-art while elaborating on certain key concepts encountered throughout the thesis and highlighting open problems. Context and contribution of each publication are summarized in Chapter 3. It demonstrates the interconnections between the published work and illustrates the broader context. Finally, Chapter 4 concludes the work by summarizing the overall contribution of the present thesis and by discussing possible future directions.

The relevant publications are attached as post-print copies at the end of the thesis. They are ordered by project and by increasing complexity.

Chapter 1

Introduction

We may assume the superiority ceteris paribus [other things being equal] of the demonstration which derives from fewer postulates or hypotheses.

— Aristoteles, *Posterior Analytics*

Large parts of our planet's landmass are inaccessible by wheeled vehicles but are nonetheless often populated by legged animals, including man. This observation has motivated engineers to study and investigate the principles of legged locomotion in order to extend the range of robotic applications and thereby allowing robots to increasingly take over dangerous, difficult, or repetitive tasks in various fields such as search and rescue, agriculture, mining, nuclear power, forestry, resource exploration, health care, or public services.

The field of legged robotics has seen a significant progress in recent years [18, 70, 101, 118]. The platforms have become more capable and robust and have reached a point where they can actually leave lab environments and carry out tasks in real-world scenarios. While these improvements have mainly been achieved on a hardware and control level, it is essential that the perceptive capabilities of the robotic platforms sustain this progress. As part of this, state estimation adopts a central role since estimated quantities are often prerequisites for other tasks such as balance control, trajectory planning, target tracking, or terrain mapping. Particularly for systems which need constant stabilization, such as dynamically balanced systems, high bandwidth estimates of the attitude and velocity of the robot are indispensable.

The strong dependency of other tasks on the estimated outputs imposes high reliability specifications on the state estimation. Missing, delayed, or bad estimates can quickly lead to failures of the robotic platform causing potential damage to the robot or its surroundings. In a first instance this motivates the use of simple sensor modalities. By employing sensors with low data processing complexity the framework is less prone to possible failures. For instance, inertial measurements require much less processing than image streams and are less affected by bad environmental conditions such as poor illumination or missing texture. Thus, a reliable state estimation methodology should include "simple" sensors in order to guarantee the quality of

the estimation output at all times. This is further motivated by eventual bandwidth specifications which do not allow for long processing times.

Inertial sensors are nowadays often available on robotic platforms and, arguably, provide a very reliable source of information. While attitude estimates can be generated from inertial measurements only [93], position or velocity estimates are very inaccurate due to the underlying numerical integration of acceleration measurements. On the other hand, legged robots are often also equipped with kinematic sensors such as joint encoders. In conjunction with the assumption that foot-ground contacts remain stationary, this offers a further source of information. The first part of this thesis, entitled *Proprioceptive State Estimation for Legged Robots* addresses this matter and focuses on the sensor fusion of inertial and kinematic data.

Relying on proprioceptive sensor modalities only has the disadvantage that there are very limited means to counteract estimation drift, i.e., the accumulation of errors over time that are not being corrected for. In cases where inertial measurements are available, this mainly concerns the position and yaw angle estimates of the robot, since inclination angles (and velocities) are typically observable. In essence, the robot is comparable to a blindfolded person which finds itself moved to an unknown location: While the individual is able to estimate its inclination w.r.t. the gravity direction and its local velocity, it has no means to gauge its location or the direction it is facing to if no additional cues are available. One such additional cue is provided by the human visual perception system with which a person is able to localize w.r.t. known landmarks. For robotic systems, analogous information can be retrieved by the means of cameras. Due to their low weight and power consumption as well as the relatively affordable pricing, many robotic systems are nowadays equipped with visual sensing.

Vision based state estimation has been the focus of a large research community in the past few years. This is probably due the maturity of the sensor devices and the rich sensory information, which allows vast algorithmic possibilities. While state-of-the-art visual localization and mapping algorithms have achieved very astonishing performance in terms of accuracy and map size, the performance quickly degrades in more challenging situations. Difficulties typically arise from conditions such as missing texture, bad illumination, motion blur, or dynamic scenes. As aforementioned, a failure of the state estimation for real-world robotic systems can lead to disastrous events and should therefore be avoided or at least detected. One possible approach to render the visual state estimation more robust is to integrate measurements from additional sensor modalities. This will be the focus of the second part of this work, entitled *Robust Visual-Inertial Sensor Fusion*, where we will investigate tight visual-inertial sensor fusion while keeping a particular focus on the robustness of the methods.

Chapter 2

State-Of-The-Art

This chapter reviews the current state-of-the-art. A detailed discussion of the related work is provided in order to illustrate the encompassing context of this thesis and to introduce certain key concepts. After a brief review on legged robotics, we present the state-of-the-art relating to both parts of this thesis.

2.1 Legged Robotics

First autonomous legged robots have emerged in the late mid 20th century [42, 94]. For a long time the focus remained on classical position controlled hardware and thus locomotion and maneuvers were often bound to slow and static motions. More recently, the emergence of dynamically stabilized robots has allowed to overcome these limitations by enabling more adaptive and versatile interaction with the environment. This has been impressively demonstrated by the Boston Dynamics robots, including the well-known quadruped BigDog [18, 111] or their humanoid counterpart Atlas.

In the scope of this work we closely worked on both ETH Zurich quadruped robots StarlETH [68] and ANYmal [70] (see Figure 2.1). Both are approximately dog-sized torque-controlled robots equipped with joint encoders, contact sensors, IMU, and a varying collection of exteroceptive sensor modalities. Some additional experiments have been performed on the hydraulically-actuated quadruped HyQ [118] and on the SARCOS humanoid [117].

2.2 Part A: Proprioceptive State Estimation for Legged Robots

In comparison to other locomotion technologies, legged robots distinguish themselves in the way they interact with their surroundings through intermittent ground contacts. This can represent an additional source of information if corresponding sensory data is available. One of the earliest approaches leveraging kinematic information was developed by Roston and Krotkov [116] on their Ambler hexapod. Using a forward

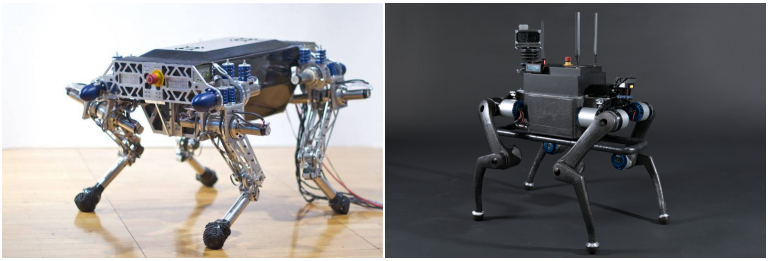

Figure 2.1: The two quadrupeds employed for the evaluation of the developed algorithms. Left: StarlETH [68]. Right: ANYmal [70].

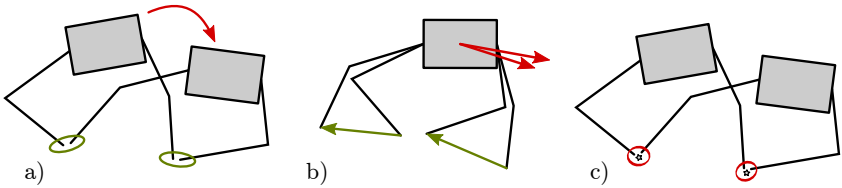

Figure 2.2: This figure illustrates three common concepts for leveraging leg kinematic information. Quantities in green are only intermediate. Quantities in red are passed to the encompassing estimation framework. Left: foothold matching generates incremental pose estimates [43, 51, 87, 116]. Middle: assuming knowledge on the incremental rotation linear velocity estimates can be derived from every foot constraint [35, 92]. Right: the "raw" constraints is directly forwarded to the encompassing framework. Here the residual is given by the error between forward kinematics and footholds [10, 11, 117].

kinematics model the location of the point feet can be computed w.r.t. the robot body. Assuming that feet in contact with the ground remain stationary, they can be matched between successive timesteps and thereby used to calculate the incremental motion. This basically minimizes the motion of the contact feet as perceived from an inertial coordinate frame and is analogous to the well-known iterative closest point method [110], whereas the point-to-point associations are already known. This concept is investigated in Paper II and a possible extension to *flat feet* is proposed. Figure 2.2 provides a coarse overview of methods for leveraging leg kinematic information.

Several research groups have extended the above foothold matching methodology. For example, Gaßmann et al. [43] introduce fuzzy weights in order to describe the reliability of the ground contact constraints. Contacts which are for instance suspected to slip are associate with a lower reliability. This is assessed by the means of various measurements, such as force and motor current measurements. The subsequent foothold matching takes this reliability score into account by down-weighting unreliable contact

constraints. Another option is to include additional sensor modalities. IMU measurements are often used in Kalman filter [77] based sensor fusion by integrating them into the prediction model [23, 35, 92]. This is a highly established approach as it allows to leverage raw IMU measurements while keeping a reasonable filter state size, avoiding additional motion models, and offering the possibility for online bias estimation. Lin et al. [87] propose such an approach for their under-actuated hexapod RHex and embed the incremental pose estimates from the foothold matching into the update step of the Kalman filter. This allows the estimation framework to bridge phases where the solution to the foothold matching is not well-defined, i.e., when less then three ground contacts are available. An additional advantage which comes with the use of inertial measurements, is that they provide information about the orientation w.r.t. the gravity direction, which, depending on the employed control framework, may be an essential asset. An alternative source of information for estimating the gravity direction can be retrieved through force measurements as these reflect the constant effort required to compensate the effect of gravity. If joint torque measurements are available, roll and pitch estimates can be acquired based on a quasi-static assumption [51]. Again, these inclination estimates can be fused with the ego-motion estimates obtained from foothold matching and can be combined with inertial data [23].

A fundamental issue with foothold matching is that a minimum of three contact points is required in order to obtain a fully-defined estimate of the incremental motion. In order to overcome this limitation for more dynamic scenarios, data-driven approaches can be used. For instance, stride length can be determined by using joint encoders, force sensors and IMU data [113]. The disadvantage here is that the state estimation needs to be trained whereas generalization to various locomotion patterns or environment types is not guaranteed. In cases where the attitude can be assumed to be known, incremental position estimates (or velocity estimates) can be obtained from a single ground contact (see Paper II for more details). This has often been applied on humanoid robots equipped with high performance IMUs which provide high quality standalone attitude estimates [35, 92].

Another option is to rely on a dynamic model and draw on the torque and force sensor measurements which are available on many modern legged robotic platforms. However, due to non-modeled effects and external disturbances as well as due to the dependency on parameters which tend to be difficult to identify (e.g. moments of inertia or friction coefficients), the overall accuracy of dynamic models tends to be limited. Furthermore, the evaluation of the equations of motion for a fully actuated robot can be computationally demanding especially if derivatives w.r.t. to state variables (e.g. robot attitude or velocities) are required. A brief sketch how to include the full dynamic model is provided in Paper II. The relatively low accuracy and the high computational costs motivate and justify the use of approximations. For instance, the increased costs emerging from the Jacobian computation can be avoided by using a decoupled steady state Kalman filter [144] or by treating the prior state as a fixed value [143]. Both approaches involve stochastic simplifications and may become inconsistent due to the negligence of cross-correlations. Other common approximations are the Spring Loaded Inverted Pendulum (SLIP) model [53], the Linear Inverted Pendulum Model (LIPM) [129, 142], or the use of two-dimensional (2D) sagital dynamics [2, 82].

A first relation to the second part of this thesis is given by the fact that a single

foothold can be interpreted as a stationary landmark in the environment (see Paper III). Modeling the world as collection of 3D landmarks has been frequently exploited in the fields of visual odometry and Simultaneous Localization And Mapping (SLAM) [28, 81, 100]. This has also been shown to be combinable with IMU data, where robust state estimation can be obtained through *tight* sensor fusion [73, 79]: This combines sensor data in a in a less-processed form and thereby leads to a stochastically more consistent formulation. Furthermore, a nonlinear observability analysis can be performed for such systems in order to show that all states are observable except for the global position and yaw angle [96]. The required motion for exciting the observable states depends on the amount of information that can be extracted from a landmark observation. In the context of legged robotics, this depends on the ground contact modeling. For instance, many humanoid robots exhibit flat feet where the additional rotational constraints around the feet provide extra information [117].

Most methods leveraging kinematic information assume stationary contact points. While a certain amount of slippage can be handled the use of an explicit slip detection and managing can be beneficial to avoid corruption of the state estimation. A relatively simple approach draws on the observation that non-slipping ground contacts maintain a constant distance w.r.t. each other. Especially for robots with larger number of legs this becomes interesting since a slipping foothold violates more than one relative distance constraint [116]. In filtering setups, the stochastic nature of the approach can be used to evaluate the probability of observing a given measurement [52]. In Paper IV we discuss a way to apply this to legged robotic state estimation in order to reject the kinematic measurements of slipping feet. If available, the consideration of further sensor readings such as force sensors can also be helpful for slip detection [78].

The use of filter-based approaches also allows for an easy inclusion of further sensor modalities. In order to reduce the accumulation of drift exteroceptive sensor modalities are of high interest. A classical approach is to include GPS data. This can easily be achieved within a filter-based approach by integrating position measurements within the update step [43]. In particular when the nature of the sensory data becomes more complex, the information is often fused in a *loose* manner, i.e., the exteroceptive data is first processed to an intermediate ego-motion estimate before it is combined with the proprioceptive measurements [23, 39, 92]. A generic approach to introduce external six-dimensional (6D) pose measurements is described in Paper II. While this improves modularity and keeps the framework manageable, it often results in a stochastically suboptimal solution. In the context of legged robotics, only Fallon et al. [35] propose a semi-tight fusion of exteroceptive sensing by computing a Light Detection And Ranging (LiDAR) based innovation term on the sub-state of the filter.

2.3 Part B: Robust Visual-Inertial Sensor Fusion

Many visual localization methods rely on the assumption that the environment can be modeled as a collection of stationary 3D landmarks. Fundamental contributions, in both visual localization [28, 81] and visual reconstruction [135], have relied on this concept. The idea is to extract salient points [56, 120] in a specific image and to find correspondences in neighboring frames. This can be either done by matching

landmark descriptors [88] or by using tracking approaches [89, 120]. The later rely on good initial guesses for the location of landmarks in the camera frames.

For 3D monocular vision, Davison [28] proposed one of the first real-time localization algorithms. It relies on an Extended Kalman Filter (EKF) based framework, where the 3D locations of the landmarks are part of the filter state and thus co-estimated online. A constant velocity model is employed as process model and the reprojection errors between predicted and measured landmark image locations serve as innovation terms in the Kalman filter update step. As both localization and mapping are performed in a unified framework, this approach can be classified as a SLAM algorithm. In subsequent contributions a trend towards batch optimization based frameworks can be observed, especially after the seminal work of Klein and Murray [81]. One reason for this was that frameworks which jointly optimize map and motion for a given number of keyframes (a selected subset of all camera frames) exhibit a much better scalability in terms of landmark count [130]. Also, due to the possibility to refine past estimates, consistency issues related to the continuous linearization and marginalization of filtering frameworks [21, 75] could be mitigated. This resulted in some impressive visual SLAM solutions [76, 95, 100, 131].

One fundamental issue with visual approaches is the projective nature of the camera models which, without any additional cues, prevents the inference of the distance of perceived landmarks from a single view. Thus, in the case of monocular vision, camera motion is essential for retrieving the 3D geometry of the scene. However, as the distance information gets gradually available with increasing baseline, it is not straightforward how to represent the corresponding estimates. For instance, frameworks with regular Cartesian coordinates cannot properly capture the one-directional initial uncertainty and thus often fall back to delayed landmark initialization schemes [28], where a landmark is only properly initialized once it has been observed from different view points. Fortunately this limitation can be overcome by using an inverse-depth parametrization of the landmark position [98]. In essence this splits the representation of the landmark location into a direction it is viewed in and its inverse depth (or distance). This partitions the state into an initially unobservable part (depth) and an observable part which can be properly initialized for a new landmark (direction). The additional inversion of the depth parametrization alleviates issues related to non-linearities and allows a more appropriate stochastic representation of the depth [126]. Note that for batch optimization based approaches regular Cartesian parametrization is often sufficient since landmarks can be initialized once observed from multiple points of view and can also be refined thereafter.

While visual odometry or SLAM approaches can achieve great results in terms of accuracy, they often struggle in more difficult scenarios involving fast motions or missing texture. A common approach to improve robustness is sensor fusion whereby vision has often been combined with inertial data due to the complementarity of both sensor modalities: IMUs provide reliable and high bandwidth incremental pose estimates but quickly suffer from drift due to the integration of sensor noise. On the other hand, visual information can be used to observe landmarks over a prolonged time period and thereby mitigate the accumulation of drift, but can potentially be prone to estimation failures. Within an EKF framework, tight fusion of both sensor modalities can be achieved by including the IMU measurements into the process model of the classical EKF SLAM approach [28]. Additional challenges are caused by the

need for online calibration of the accelerometer and gyroscope biases which are often modeled as slowly time varying quantities [73, 79].

In the context of visual-inertial sensor fusion the superiority of batch optimization based approaches is less evident. This may be due to the more accurate incremental motion estimates which alleviate problems caused by the EKF linearization as the deviation from the true trajectory is less severe. Also, in order to integrate IMU measurement in a batch optimization framework the temporal distance between keyframes cannot exceed a certain value and thus the number of required keyframes increases with time. This in turn impairs the scalability of these frameworks and often marginalization of parts of the map are unavoidable for maintaining real-time capability [84]. Furthermore, methods for mitigating consistency issues related to the spurious observability of unobservable states can also improve the consistency of filter-based approaches. The two most common ones are partitioning of observable and unobservable states [21] or the numerical enforcement of unobservability constraints [61]. Due to its simplicity we will follow the first approach in Paper V, Paper VI, and Paper VII. Another option to improve consistency while at the same time reducing computational costs is to follow a structureless approach where the landmark locations are directly marginalized during construction of the reprojection error. This has been embedded in both a filter-based framework [99] or batch optimization [41].

A recent and popular trend is the transition to direct photometric approaches. Instead of relying on the tracking of 3D landmarks these methods directly minimize the error between predicted/rendered and measured image regions, and are consequently often referred to as *direct* methods. One option is to model the environment as a collection of planar image patches and derive a photometric residual capturing geometry (location/normal of patches), camera motion, as well as illumination changes [71, 97, 122]. The classical KLT-tracker [120] is related to those approaches and can be seen as direct method for landmark tracking. In Paper VI and Paper VII the concept of direct patch-based method is combined with an IMU-driven EKF in a fully robot-centric setup. Tanskanen et al. [132] present a similar method but deviate in how they parameterize the patch location. An alternative is to compute a per-pixel estimate of the depth for the partial or full image. This can be achieved within keyframe based batch optimization where the algorithm alternates between depth estimation and direct image alignment [33]. This has also recently been combined with inertial data by formulating a joint energy term composed of visual and inertial residuals [136].

Chapter 3

Contribution

In this chapter we discuss contexts and contributions of publications included in this thesis. We also highlight the relations between the different papers and show how they fit into a coherent research framework. A recurring link throughout this work is given by the underlying methods and tools, e.g., sensor fusion methods, parametrization and modeling tools, or nonlinear observability analysis. Paper I reviews the proper handling of 3D orientations in the context of optimization frameworks, which is a key component in mobile robotic state estimation. Thereafter the papers are ordered by project and by increasing complexity.

3.1 Background Theory

The paper included in this section discusses the representation and handling of 3D orientations, which is a central part of this thesis. This is due to the high occurrence of optimization based methods in state estimation and the frequent involvement of 3D orientations. However, the scope of its applicability is much broader than discussed here, and includes every optimization based technique involving 3D orientations. In robotics, notable examples include motion control or path planning.

Paper I

Michael Bloesch, Hannes Sommer, Tristan Laidlow, Michael Burri, Gabriel Nuetzi, Péter Fankhauser, Dario Bellicoso, Christian Gehring, Stefan Leutenegger, Marco Hutter, Roland Siegwart, "A Primer on the Differential Calculus of 3D Orientations". *CoRR*, arXiv.org 2016.

Context

Many filtering and optimization problems in engineering require the representation of 3D orientations. A classical example is the orientation of free floating base robots

such as legged robots or micro aerial vehicles. Issues typically arise when 3D orientations become part of the free variables that are being optimized for. This is caused by the fact that the set of 3D orientations is not a vector space and thus classical gradient based approaches will not work as usual. Unfortunately, conceptually simple but suboptimal solutions are often employed. These may include singularity-affected representations or over-parametrization. Furthermore, these solutions can be computationally more expensive due to complex analytical derivatives, larger state spaces, and slower convergence.

Contribution

The contribution in this paper consists in the condensation of differential geometric concepts for 3D orientations and improving the accessibility for engineers. A more abstract notion of 3D orientations is conveyed which helps avoiding issues related to different conventions. A simple notation is introduced and used to construct boxplus and boxminus operators which adopt the roles of addition and subtraction on the Lie group $SO(3)$. Combining this with a regular definition of differentials then leads to simple and minimal analytical derivatives of quantities involving 3D orientations. Exemplification is provided in the form of a commonly encountered filtering problem and through various derivations and proofs. The presented derivation for the Jacobian of the exponential map is of special interest as it is much more compact when compared to the standard series expansion based proof.

Interrelations

The concepts presented in this paper are used throughout the entire thesis and play a central role within the other contributions. Many results achieved in this thesis strongly rely on a proper handling of 3D orientations in order to simplify the implementation and to improve consistency and accuracy of the state estimation methods. The concepts and notations have evolved during the course of the thesis and thus some of the publications are based on differing versions.

In Paper VI and Paper VII a similar concept is derived for the representation of bearing vectors on the 2D unit sphere.

3.2 Part A: Proprioceptive State Estimation for Legged Robots

In this section we include the contributions related to the proprioceptive state estimation for legged robots. While the first paper provides an overview of the state-of-the-art and elaborates on basic concepts, the two subsequent papers offer theoretical contributions and include a detailed discussion thereof.

Paper II

Michael Bloesch, Marco Hutter, "Technical Implementations of the Sense of Balance". *Humanoid Robotics: a Reference*, Springer 2017.

Context

The sense of balance is an essential component of legged robotic systems. Using the available sensor readings a legged robot must be able to infer its posture and motion. Kinematic encoders, force sensors, or IMUs can be among the employed sensor modalities whereby many different approaches have been proposed for leveraging the information contained in the corresponding sensor readings. While methods can rely on a single sensor modality only, multiple sensor readings can also be combined or fused. In the context of legged robotics, kinematic sensors are of high interest as they represent a mean to extract information from the intermittent foot-ground contacts.

Contribution

The paper summarizes the basic concepts related to legged state estimation. This includes kinematic and dynamic modeling of legged robots, as well as basics in sensor modeling and sensor fusion. In addition to reviewing many state-of-the-art state estimation approaches, more detailed and partially novel considerations are provided for incremental foothold matching during statically stable locomotion patterns. These allow the estimation of a legged robot's ego-motion based on kinematic measurements only. Methods for *point foot* contacts and *flat foot* contacts are discussed, whereby closed-form solutions to the incremental motion estimation problem are derived for both cases. Finally, possible methods for integrating inertial measurements, dynamic quantities, or including pose estimates from other estimation processes (such as from exteroceptive sensing) are elaborated on.

Interrelations

This paper provides a general overview of legged state estimation and puts the concepts of Paper III and Paper IV into a broader perspective.

Paper III

Michael Bloesch, Marco Hutter, Mark A. Hoepflinger, Stefan Leutenegger, Christian Gehring, C David Remy, Roland Siegwart, "State Estimation for Legged Robots – Consistent Fusion of Leg Kinematics and IMU". In *Robotics Science and Systems Conference*, 2012.

Context

A reliable state estimation without the need for restrictive assumptions such as a flat terrain or a pre-defined gait pattern is of high importance for versatile legged robots. Two sensor modalities which can potentially contribute to this goal while at the same time being available on many legged robots are IMUs and kinematic sensors. So far however, most existing approaches do not fully leverage the information contained in these measurements or rely on restrictive assumptions.

Contribution

The contribution of the presented method consists in integrating the information from the intermittent ground contacts within a state estimation framework. Inspired by the current SLAM literature, footholds are modeled as 3D stationary landmarks and co-estimated within an IMU-driven EKF. Based on a forward kinematics model, kinematic measurements can be turned into relative constraints between the estimated footholds and the robot's ego-motion. These constraints are integrated as innovation terms during the update step of the EKF. An observability analysis is performed in order to show that for a non-degenerate motions only the global position and yaw angle are unobservable, even if only a single foot is in contact with the ground. Based on Huang et al. [61], observability constraints are enforced in order to guarantee consistency of the framework. The resulting proprioceptive estimation framework achieves very high bandwidth and locally accurate state estimates and can thus be employed for stabilizing feedback control on the quadrupedal robot StarlETH [68].

Interrelations

Paper IV revisits the concept introduced in this paper and presents potential improvements. Also, the concept is extended to legged robots with flat feet where additional information can be gained from the rotational constraints [117]. Furthermore, a batch optimization based calibration routine is developed in order to estimate the kinematic parameters of the robot during locomotion [12].

Paper IV

Michael Bloesch, Christian Gehring, Peter Fankhauser, Marco Hutter, Mark A. Hoepflinger, Roland Siegwart, "State Estimation for Legged Robots on Unstable and Slippery Terrain". In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

Context

In the context of legged robotics the knowledge about the state of the foot contacts can be valuable for the control framework. For instance if a foot is detected to slip, the control framework can take this into account and take corrective actions such as decreasing the tangential force. On the other hand, a more detailed characterization of the contact state can also improve the state estimation as slipping feet could be down-weighted or even be discarded. Arguably this may be a very important property, as the combination of slippage and simultaneous degradation of the state estimation quality due to erroneous contact constraints could exacerbate the situation. However, self-contained slippage detection sensors applicable to legged robots have not been invented yet and are constrained by low payload specifications at the foot. Consequently, a sensor fusion based slip detection approach relying on more readily available sensor readings while considering the full robot model and state is desirable.

Contribution

In this paper the kinematic constraints employed in Paper III are used to derive a foothold stationarity constraint on the velocity level. This has the advantage that the filter state can be reduced since the footholds need not to be co-estimated anymore. Furthermore, due to the stochastic nature of Kalman filters, an estimate of the uncertainty of the innovation term is available and a Mahalanobis distance based test can be performed for outlier detection [52]. Thus, whenever the innovation term (which directly results from the zero velocity constraint at the foothold) exceeds a certain threshold, the corresponding foot is marked as slipping. In this event, the innovation term can be discarded and, consequently, erroneous contact constraints do not corrupt the estimation process. Additionally, an observability analysis of the system is performed in order to show that all states except for global position and yaw angle are observable. In comparison to prior work, the presented observability analysis takes into account the special nature of 3D rotations, leading to an analytically simpler observability matrix.

An adapted implementation of the filter presented in this paper has become an integral part of both quadrupedal platforms StarlETH [68] and ANYmal [70]. Further experiments have been performed on the HyQ robot [118].

Interrelations

A modular extension has been proposed where the state estimation is enhanced with external 6D pose measurements [39].

3.3 Part B: Robust Visual-Inertial Sensor Fusion

This section lists and summarizes publications related to robust visual-inertial sensor fusion. All papers present filter-based approaches and pay special attention to the robustness of the localization. In contrast to the first paper which draws on an optical flow inspired approach, the two subsequent papers investigate the use of photometric error as direct source of information.

Paper V

Michael Bloesch, Sammy Omari, Peter Fankhauser, Hannes Sommer, Christian Gehring, Jemin Hwangbo, Mark A. Hoepflinger, Marco Hutter, Roland Siegwart, "Fusion of Optical Flow and Inertial Measurements for Robust Egomotion Estimation". In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

Context

Robustness is an important property to consider for visual-inertial sensor fusion, especially if the estimates are used for real-time control of a robotic platform. Unfortunately, due to the rather difficult quantification of the robustness of localization systems, the state estimation community has often employed the error over traveled distance as primary endpoint, and consequently focused on high-accuracy localization.

Classical challenges encountered during visual-inertial localization are fast motions, short feature tracks, image blur, missing texture, lighting changes, or dynamic scenes.

Contribution

This paper deals with a first filter-based attempt for developing a robust ego-motion estimation based on the combination of inertial and visual measurements. The use of optical flow allows the extraction of visual information from a single visual match. Thus, a feature does not need to be tracked over an extended time period in order to contribute to the estimation. Furthermore, this also avoids complex initialization procedures of landmarks and simplifies the overall initialization of the state estimation system. By co-estimating the mean scene depth, a new residual can be derived for each optical flow measurement. These residuals serve as innovation term in the update step of an Unscented Kalman Filter (UKF) with an IMU-driven process model. Optical flow measurements are thereby tightly fused with inertial data leading to a localization framework which is highly robust to fast motions and short feature tracks.

Interrelations

This paper represents preliminary work towards filter-based robot-centric visual-inertial sensor fusion. Paper VI and Paper VII can be perceived as continuation of this work.

Paper VI

Michael Bloesch, Sammy Omari, Marco Hutter, Roland Siegwart, "Robust Visual Inertial Odometry Using a Direct EKF-Based Approach". In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

Context

Direct approaches attempt to leverage the photometric error directly into the estimation framework. This can have the advantage that a statistically more sound measurement model can be applied. However, as discussed and shown in previous work [71, 97, 122], a generative photometric model can be highly nonlinear and thus prone to local minima. Consequently, a good initial guess is essential for optimization based algorithms to work properly. The combination with inertial data can significantly improve this aspect whereas only the image depth remains as highly uncertain initial quantity.

Contribution

This paper tightly combines photometric residuals with inertial measurements within an EKF. This is achieved by associating every landmark with an image patch and by employing the pixel intensity error between the original patch and its projection into subsequent images as innovation term. Analogously to classical IMU-driven Kalman filters, the inertial measurements are integrated within the process model. By employing a fully robot-centric formulation of the filter state and by partitioning the

landmark representation into bearing vector and distance parameter, issues linked to spurious observability can be mitigated. This also allows an undelayed initialization of landmark locations: While the initial bearing vector can directly be guessed from the landmark's pixel coordinates the uncertainty of the corresponding distance parameter is initialized to a large value. Thus, similar to the framework in Paper V, visual information can be integrated from a landmark's second observation onwards. Online calibration of IMU biases and IMU-camera extrinsics is integrated into the open-source available software framework[1].

Interrelations

This paper employs the same IMU-driven process model as Paper V but directly leverages photometric residuals within the visual measurement model.

Paper VII

Michael Bloesch, Michael Burri, Sammy Omari, Marco Hutter, Roland Siegwart, "IEKF-based Visual-Inertial Odometry using Direct Photometric Feedback". *International Journal of Robotics Research*, SAGE 2017.

Context

In classical landmark based visual localization systems the set of employed landmarks needs to be registered w.r.t. the camera images. This data association process, whether done by extraction and descriptor matching [88] or direct tracking [120], is prone to various failure modes. A possible way to improve the data association is to include further information such as inertial data. For tracking based approaches, this can either be used to generate an initial guess or may be more tightly integrated as prior.

Contribution

In comparison to Paper VI, an IEKF is applied which has the advantage of overcoming the nonlinearities present in the generative photometric model. At the same time, this allows to omit additional data association since this process is inherently handled by the filter's iterative update step which aligns the patches within the current image. Furthermore, the inertial measurements, which are processed during the prediction step, are tightly integrated into the data association in form of the prior state estimate. This results in a visual-inertial framework which is no longer limited to the tracking of corner features, but can also track other visual features such as lines.

Additionally, this paper provides a more detailed discussion on the parametrization of bearing vectors together with the minimal representation of uncertainties. This includes insights into the differentiation of non-vector space quantities and the required adaptations on the filter level.

[1] https://github.com/ethz-asl/rovio

Interrelations

This paper extends the work of Paper VI and investigates data association and multi-camera setups while providing more insights into the mathematical derivations.

3.4 List of Publications

First author publications (sorted by year)

- M. Bloesch, S. Weiss, D. Scaramuzza, and R. Y. Siegwart. Vision Based MAV Navigation in Unknown and Unstructured Environments. In *IEEE International Conference on Robotics and Automation*, 2010. doi: 10.3929/ethz-a-010137518

- M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart. State Estimation for Legged Robots - Consistent Fusion of Leg Kinematics and IMU. In *Robotics Science and Systems Conference*, 2012. doi: 10.15607/RSS.2012.VIII.003

- M. Bloesch, M. Hutter, C. Gehring, M. A. Hoepflinger, and R. Siegwart. Kinematic Batch Calibration for Legged Robots. In *IEEE International Conference on Robotics and Automation*, 2013. doi: 10.1109/ICRA.2013.6630924

- M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger, and R. Siegwart. State Estimation for Legged Robots on Unstable and Slippery Terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013. doi: 10.1109/IROS.2013.6697236

- M. Bloesch, S. Omari, P. Fankhauser, H. Sommer, C. Gehring, J. Hwangbo, M. A. Hoepflinger, M. Hutter, and R. Siegwart. Fusion of Optical Flow and Inertial Measurements for Robust Egomotion Estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014. doi: 10.3929/ethz-a-010184819

- M. Bloesch, S. Omari, M. Hutter, and R. Siegwart. Robust Visual Inertial Odometry Using a Direct EKF-Based Approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015. doi: 10.3929/ethz-a-010566547

- M. Bloesch, H. Sommer, T. Laidlow, M. Burri, G. Nützi, P. Fankhauser, D. Bellicoso, C. Gehring, S. Leutenegger, M. Hutter, and R. Siegwart. A Primer on the Differential Calculus of 3D Orientations. *CoRR*, abs/1606.0, 2016

- M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart. IEKF-based Visual-Inertial Odometry using Direct Photometric Feedback. *International Journal of Robotics Research*, (conditionally accepted), 2017

- M. Bloesch and M. Hutter. Technical Implementations of the Sense of Balance. In *Humanoid Robotics: a Reference*, chapter HB. 2017

Co-authored publications (sorted by author and year)

- F. Bloechliger, M. Bloesch, P. Fankhauser, M. Hutter, and R. Siegwart. Foot-Eye Calibration of Legged Robot Kinematics. In *International Conference on Climbing and Walking Robot*, 2016. doi: 10.3929/ethz-a-010655381
- M. Burri, M. Bloesch, D. Schindler, I. Gilitschenski, Z. Taylor, and R. Siegwart. Generalized Information Filtering for MAV Parameter Estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016. doi: 10.1109/IROS.2016.7759483
- P. Fankhauser, M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, and R. Siegwart. Reinforcement Learning of Single Legged Locomotion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013. doi: 10.3929/ethz-a-010018685
- P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart. Robot-Centric Elevation Mapping With Uncertainty Estimates. In *International Conference on Climbing and Walking Robot*, 2014. doi: 10.3929/ethz-a-010173654
- P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart. Kinect v2 for Mobile Robot Navigation: Evaluation and Modeling. In *International Conference on Advanced Robotics*, 2015. doi: 10.3929/ethz-a-010513824
- P. Fankhauser, M. Bloesch, P. A. Krüsi, R. Diethelm, M. Wermelinger, T. Schneider, M. T. Dymczyk, M. Hutter, and R. Siegwart. Collaborative Navigation for Flying and Walking Robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016. doi: 10.3929/ethz-a-010687710
- C. Gehring, S. Coros, M. Hutter, M. Bloesch, M. A. Hoepflinger, and R. Siegwart. Control of Dynamic Gaits for a Quadrupedal Robot. In *IEEE International Conference on Robotics and Automation*, 2013. doi: 10.3929/ethz-a-010023052
- C. Gehring, S. Coros, M. Hutter, M. Bloesch, P. Fankhauser, M. A. Hoepflinger, and R. Y. Siegwart. Towards Automatic Discovery of Agile Gaits for Quadrupedal Robots. In *IEEE International Conference on Robotics and Automation*, 2014. doi: 10.3929/ethz-a-010183016
- C. Gehring, C. Bellicoso, S. Coros, M. Bloesch, P. Fankhauser, M. Hutter, and R. Siegwart. Dynamic Trotting on Slopes for Quadrupedal Robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015. doi: 10.3929/ethz-a-010535713
- C. Gehring, S. Coros, M. Hutler, C. Dario Bellicoso, H. Heijnen, R. Diethelm, M. Bloesch, P. Fankhauser, J. Hwangbo, M. Hoepflinger, and R. Siegwart. Practice Makes Perfect: An Optimization-Based Approach to Controlling Agile Motions for a Quadruped Robot. *IEEE Robotics and Automation Magazine*, 23(1), 2016. doi: 10.1109/MRA.2015.2505910

- A. Handa, M. Bloesch, V. Patraucean, S. Stent, J. McCormac, and A. Davison. Gvnn: Neural Network Library for Geometric Computer Vision. In *European Conference on Computer Vision*, 2016. doi: 10.1007/978-3-319-49409-8_9

- L. Hertig, D. Schindler, M. Bloesch, C. D. Remy, and R. Siegwart. Unified State Estimation for a Ballbot. In *IEEE International Conference on Robotics and Automation*, 2013. doi: 10.1109/ICRA.2013.6630913

- M. A. Hoepflinger, M. Hutter, C. Gehring, M. Bloesch, and R. Siegwart. Unsupervised Identification and Prediction of Foothold Robustness. In *IEEE International Conference on Robotics and Automation*, 2013. doi: 10.1109/ICRA.2013.6631036

- M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, C. D. Remy, and R. Siegwart. StarlETH: A Compliant Quadrupedal Robot for Fast, Efficient, and Versatile Locomotion. In *International Conference on Climbing and Walking Robot*, 2012. doi: 10.3929/ethz-a-010034688

- M. Hutter, M. A. Hoepflinger, C. Gehring, M. Bloesch, C. D. Remy, and R. Siegwart. Hybrid Operational Space Control for Compliant Legged Systems. In *Robotics Science and Systems Conference*, 2012. doi: 10.3929/ethz-a-010184796

- M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, and R. Y. Siegwart. Walking and Running with StarlETH. In *International Symposium on Adaptive Motion of Animals and Machines*, 2013. doi: 10.3929/ethz-a-010022793

- M. Hutter, M. Bloesch, J. Buchli, C. Semini, S. Bazeille, L. Righetti, and J. Bohg. AGILITY – Dynamic Full Body Locomotion and Manipulation with Autonomous Legged Robots. In *IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2013. doi: 10.3929/ethz-a-009996472

- M. Hutter, H. Sommer, C. Gehring, M. A. Hoepflinger, M. Bloesch, and R. Siegwart. Quadrupedal Locomotion using Hierarchical Operational Space Control. *The International Journal of Robotics Research*, 33(8), 2014. doi: 10.3929/ethz-a-010184871

- M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, P. Fankhauser, and R. Y. Siegwart. Excitation and Stabilization of Passive Dynamics in Locomotion using Hierarchical Operational Space Control. In *IEEE International Conference on Robotics and Automation*, 2014. doi: 10.3929/ethz-a-010184874

- M. Hutter, C. Gehring, M. A. Hoepflinger, M. Bloesch, and R. Siegwart. Toward Combining Speed, Efficiency, Versatility, and Robustness in an Autonomous Quadruped. *IEEE Transactions on Robotics*, 30(6), 2014. doi: 10.1109/TRO.2014.2360493

- M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann,

A. Melzer, and M. Hoepflinger. ANYmal – a Highly Mobile and Dynamic Quadrupedal Robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016. doi: 10.3929/ethz-a-010686165

- M. Neunert, M. Bloesch, and J. Buchli. An Open Source, Fiducial Based, Visual-Inertial Motion Capture System. *International Conference on Information Fusion*, 2015

- S. Omari, M. Bloesch, P. Gohl, and R. Siegwart. Dense Visual-Inertial Navigation System for Mobile Robots. In *IEEE International Conference on Robotics and Automation*, 2015. doi: 10.1109/ICRA.2015.7139554

- C. D. Remy, M. Hutter, M. A. Hoepflinger, M. Bloesch, C. Gehring, and R. Siegwart. Quadrupedal Robots with Stiff and Compliant Actuation. *Automatisierungstechnik*, 60(11), 2012. doi: 10.3929/ethz-a-010000217

- N. Rotella, M. Bloesch, L. Righetti, and S. Schaal. State Estimation for a Humanoid Robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014. doi: 10.1109/IROS.2014.6942674

- L. Wagner, P. Fankhauser, M. Bloesch, and M. Hutter. Foot Contact Estimation for Legged Robots in Rough Terrain. In *International Conference on Climbing and Walking Robot*, 2016. doi: 10.3929/ethz-a-010643823

3.5 List of Supervised Students

The following students and theses have been supervised during the author's doctoral studies:

Master Thesis

Master student, 6 months full time

- Timothy Sandy (Spring 2013): "Localization and Mapping Using a Laser Range Finder and Inertial Measurement Unit"

- Christoph Müri (Fall 2013): "Fusion of Optical Flow and Inertial Measurements for Egomotion Estimation"

- Rico Kreis (Spring 2014): "State Estimation for a Quadruped Robot using Force Sensors"

- Dominik Schindler (Spring 2015): "Visual Terrain Estimation for Legged Robots"

- Gautham Manoharan (Spring 2015): "Terrain Estimation Using Range Measurements for Legged Robots"

- Anna-Maria Georgarakis (Fall 2015): "Comparison of Optimization Methods for Legged Motion Synthesis"
- Frederike Dümbgen (Spring 2016): "Visual Terrain Estimation for Legged Robots"

Semester Thesis

Master student, 3-4 months part time

- Marius Fehr (Spring 2014): "Monocular Dense 3D Reconstruction"
- Manuela Eugster (Spring 2014): "Control Strategy for a Quadrupedal Robot on Slippery Ground"
- Martin Wermelinger (Spring 2014): "Stability Checking for Legged Rough Terrain Locomotion"
- Gabriel Hottiger (Spring 2014): "High Quality Terrain Scanning"
- Kaiser Fabian (Spring 2014): "Joint State Estimation of Flying and Walking Robots"
- Diego Rodriguez (Fall 2014): "Evaluation and Characterization of the Kinect V2 Sensor" [38]
- Basil Weibel (Fall 2014): "Characterisation of Fiducials for Visual Take-off and Landing of Aerial Robots"
- Lucas Wagner (Fall 2015): "Foot Contact Estimation for Legged Robots in Rough Terrain" [138]
- Benjamin Müller (Fall 2015): "Tip Over Stability for Legged Robots or Walking Excavators"
- Fabian Blöchliger (Fall 2015): "Visual Calibration of Legged Robot Kinematics" [7]
- Fabian Tresoldi (Fall 2016): "State Estimation of a Walking Excavator"

Bachelor Thesis

Bachelor student, 3-4 months part time

- Dominik Schindler and Hertig Lionel (Spring 2012): "Development of an Extended Kalman Filter for Ballbot Rezero" [58]
- Daniel Singer and Sonja Segmüller (Spring 2012): "Reflexes on a Spider Robot"
- Roman Ebneter and Lucas Eicher (Spring 2012): "Gait Pattern Optimization for a Spider Robot"

- Ralph Aeschimann (Spring 2013): "Modeling and System Identification of a Quadrotor Helicopter"
- Milan Schilling (Spring 2015): "Visual Recognition of Stairs for a Stair Climbing Wheelchair"

Studies on Mechatronics

Bachelor student, literature review, 3-4 months part time

- Felix Renaut (Fall 2012): "MEMS Intertial Sensor Technology"

Chapter 4

Conclusion and Outlook

In this chapter the main results are summarized and prospective research directions are outlined.

4.1 Part A: Proprioceptive State Estimation for Legged Robots

Research presented in the first part of this thesis focuses on providing reliable and high-bandwidth ego-motion estimates for legged robots. A key contribution is the extraction of information from single ground contacts without relying on restrictive assumptions such as a horizontal terrain or a specific gait pattern. Two methods have been proposed to this end: The first method co-estimates the location of footholds and employs the kinematic data as relative measurement between foothold and main body. The second method directly defines the innovation term through the residual velocity at the foothold (which depends on both kinematic measurements and ego-motion estimates). A nonlinear observability analysis endorses the presented approaches from a theoretical side, by showing that all local control relevant states are fully observable given enough motion. The approaches have been tested on real quadrupedal robots and constitute an inherent part of the current software framework on StarlETH and ANYmal.

While a concept to include exteroceptive sensing in a modular way has been discussed and is in use on the actual robots, a thorough evaluation of this approach is still pending. A tighter integration of exteroceptive sensing could also be conceivable if modularity is not an essential specification. This could lead to a more robust estimation framework as the exteroceptive data processing and state estimation could benefit from the proprioceptive sensing. For instance, a classical filter-based visual SLAM algorithm could be enhanced with inertial and kinematic measurements. In case the robot moves into a texture-less region, the state estimation would inherently rely more on the proprioceptive sensing and thus avoid divergence due to the lack of visual inputs. The disadvantages of this approach are an increased complexity and a

loss in modularity as the estimation framework is tailored to a specific exteroceptive sensing modality.

Another only briefly discussed topic is the possible integration of dynamic sensors and models. While they generally exhibit a higher noise and uncertainty magnitude, they could serve as redundant source of information and for instance help to detect and amortize IMU sensor saturation (in certain setups this can occur frequently). At the same time this could also be used to co-estimate certain dynamic quantities and thereby refine the quality of the dynamic model employed by the control algorithms.

A further related research direction is the better characterization of contact situations, including estimation of the contact force, the contact normal, or the contact velocity. Some of the mentioned quantities are not observable in general scenarios, but the contact force should for instance be retrievable through the use of a dynamic model together with the ego-motion estimates and the joint torques. Alternatively, using additional sensor modalities (see [138]) or making sensible assumptions on the setup may lead to the estimation of further quantities, such as payload mass or external forces. Eventually, also situations where other body parts, e.g. the shank or the knee, are in contact with the environment could be detected and specified.

4.2 Part B: Robust Visual-Inertial Sensor Fusion

The lack of long-term accuracy of proprioception-only approaches was an incentive for investigating exteroceptive sensing capabilities. Due to the inherent complementarity of both sensor modalities we investigated the application of visual-inertial sensor fusion. Special attention was payed to the robustness of the developed frameworks. This motivated the use of simple and manageable estimation approaches with straightforward and unconstrained initialization procedures.

The bearing vector and distance parameter partitioning of the robot-centric landmark representation allows an undelayed landmark initialization, where information gets extracted out of a landmark's second observation onwards. The framework directly starts tracking landmarks as they appear in the field of view of the camera and, consequently, there is no need for a cumbersome system initialization routine. This enables an almost arbitrary resetting of the filter as long as the initial states are within a reasonable range (this is covariance tuning dependent). At the same time the robot-centric landmark location partitioning mitigates observability related issues and reduces numerical inaccuracies. Finally, by employing a singularity-free parametrization together with a minimal representation of differences and derivatives, both for 3D orientations and bearing vectors, compact and numerically efficient filter equations can be derived.

The attained level of "tightness" in terms of fusion between visual and inertial data has not been demonstrated previously. To the best of our knowledge and along with Tanskanen et al. [132], this work presents one of the first frameworks that combines inertial and photometric data within a unified localization and mapping framework. Other approaches either pre-process the photometric data or do not fuse it with inertial measurements. Note that the presented framework estimates structure (patch locations) and motion conjointly but does not yet co-estimate the texture (certain cross-correlations are thus ignored in the filter). A full joint optimization of motion,

structure, and texture may be part of future research. Another key contribution is that the data association is inherently taken care of by the filter, whereby both photometric and inertial data are considered. This extends the set of tracked image features beyond mere corner features whereby line-shaped visual structures can be leveraged as well.

While robustness has been achieved in the presence of fast motion, situations with lack of motion remain a problem for the presented monocular visual-inertial approach. Due to the projective nature of camera models the distance of the scene cannot be inferred if no motion is present. Furthermore, as the estimation of the linear velocity strongly depends on landmark distance estimates, an increasing uncertainty of the distance will lead to an increasing uncertainty of the linear velocity. This can put the estimation process in a fragile state where it is susceptible to external disturbance or outliers (e.g. moving objects). Various adaptations could alleviate this issue. Preliminary tests with zero-velocity updates, applied when no motion is apparent, have been carried out within the current framework, but the results remain to be evaluated. The exploration of other virtual updates, hybrid models, or novel feature parametrization (for instance allowing a better representation of infinitely distanced features) could be directions for future work. Another option is the above mentioned integration of robot specific sensors. For instance on a legged robot, the kinematic measurements could be directly integrated into the same estimation framework. This could prevent divergence of the filter state if the robot remains stationary or in surroundings with lack of visual features.

The use of semantic information could be a further option for increasing robustness. In the simplest setup, past tracking information could be linked to the corresponding image patch content and used to train a learning algorithm to predict the quality of candidate landmarks. This could also be taken a step further, where for instance an image classification algorithm is applied on the visual inputs. Depending on the determined class a "visual reliability" measure could be derived and used to mitigate the effect of unmodeled disturbances such as reflections or moving objects.

A well-known limitation of filter-based approaches is the bad scalability w.r.t. increasing state size. This strongly limits the number of features that can be tracked and therefore the integration of a back-end mapping represents an interesting and useful extension. Again, there are many different options to achieve this. Probably one of the simplest options is the combination of the presented visual-inertial odometry with a bundle adjustment based mapping framework where map landmarks are fed back in a loose manner to the odometry [90]. Alternatively, a batch optimization formulation of the framework could be considered. However, a robot-centric representation is not well suited in batch optimization frameworks as it leads to the addition of per-landmark per-keyframe states if implemented in a naive way. Still, some concepts, such as the photometric patch based residuals, could be transfered into a batch optimization framework.

Some last comments concern the modeling of the environment as collection of flat patches, which, depending on the actual surroundings, can induce significant modeling inaccuracies. Several improvements are imaginable: One option is to enhance the model in an offline manner and for instance allow for more complicated geometric shapes. This could involve collecting measurements for a given patch and optimize its shape and texture such that the localization process, which is running in parallel, can

make use of a more accurate photometric error model. Alternatively, additional shape parameters (e.g. the patch normal or a curvature parameter) could be tightly included into the estimation framework. Statistically, this represents the more sound method, but also involves augmenting the filter state which induces increased computational costs and consequently must be limited to few additional degrees of freedom. For instance, adding the rotation angle of the patch is an interesting option. In comparison to the two traditional measurement quantities, the x- and y-coordinates of a landmark, this adds a third degree of information around the viewing axis of the landmark. From an observability point of view this increases the amount of information gained from a single landmark observation and thereby could increase accuracy (more information) and robustness (less landmarks required, less unobservable modes) of the framework.

Paper

A Primer on the Differential Calculus of 3D Orientations

Michael Bloesch, Hannes Sommer, Tristan Laidlow, Michael Burri, Gabriel Nuetzi, Péter Fankhauser, Dario Bellicoso, Christian Gehring, Stefan Leutenegger, Marco Hutter, Roland Siegwart

Abstract

The proper handling of 3D orientations is a central element in many optimization problems in engineering. Unfortunately many researchers and engineers struggle with the formulation of such problems and often fall back to suboptimal solutions. The existence of many different conventions further complicates this issue, especially when interfacing multiple differing implementations. This document discusses an alternative approach which makes use of a more abstract notion of 3D orientations. The relative orientation between two coordinate systems is primarily identified by the coordinate mapping it induces. This is combined with the standard exponential map in order to introduce representation-independent and minimal differentials, which are very convenient in optimization based methods.

Published in:
CoRR, arXiv.org 2016
http://arxiv.org/abs/1606.05285

1 Introduction

The primary goal of this document is to derive and summarize the most important identities for handling 3D orientations. It can readily be used as a look-up document (general identities are green (Section 4), implementation dependent identities are red (Section 5)). In a compact theoretical section all equations are derived together with some insights into their mathematical background (Section 3). We believe however, that the best way to understand these concepts is to apply the presented findings on an actual system. To this end, we discuss the modeling of an Inertial Measurement Unit (IMU) driven kinematic model (Section 6). Furthermore, we provide the most important proofs and derivations in order to provide some additional insights and examples. Similar elaborations on the topic exist in [4, 30, 31].

An understanding of kinematics (including the concept of coordinate systems) is a prerequisite for understanding this document. The corresponding conventions and notations are summarized in Section 2. To completely follow the theoretical sections some higher mathematical concepts are necessary.

2 Vectors and Coordinate Systems Notation

In this document coordinate systems are denoted by calligraphic capital letters, e.g. $\mathcal{A}$, and coordinate tuples are represented by bold lower case letters, e.g. ${}_{\mathcal{A}}\boldsymbol{r}_{\mathcal{BC}}$. The left-hand subscript of a coordinate tuple indicates the coordinate system the vector is represented in, while the right-hand subscripts indicate the 3D points related to start and end points. For instance, the term ${}_{\mathcal{A}}\boldsymbol{r}_{\mathcal{BC}}$ denotes the coordinates of a vector $\vec{\boldsymbol{r}}_{\mathcal{BC}}$ (denoted with an arrow) in the Euclidean space $\mathbb{E}^3$ from point $\mathcal{B}$ to point $\mathcal{C}$, represented in the coordinate system $\mathcal{A}$. By abuse of notation, we denote the origin associated with a specific coordinate system by the same symbol. Furthermore, the term $\Phi_{\mathcal{BA}} \in SO(3)$ is employed for representing the relative orientation of a coordinate system $\mathcal{B}$ w.r.t. a coordinate system $\mathcal{A}$. Its definition is coupled to the (distance preserving) mapping of coordinate tuples and we employ the notation ${}_{\mathcal{B}}\boldsymbol{r}_{\mathcal{BC}} = \Phi_{\mathcal{BA}}\,({}_{\mathcal{A}}\boldsymbol{r}_{\mathcal{BC}})$. We define the mapping $\boldsymbol{C} : SO(3) \to \mathbb{R}^{3\times3}$ such that $\Phi(\boldsymbol{r}) \triangleq \boldsymbol{C}(\Phi)\boldsymbol{r}$ (corresponding to the rotation matrix). A more complete overview of coordinate systems and rotations is given in [4].

Furthermore, the vectors $\vec{\boldsymbol{v}}_{\mathcal{B}}$ and $\vec{\boldsymbol{a}}_{\mathcal{B}}$ denote the absolute (w.r.t. an inertial coordinate system) velocity and acceleration of the point $\mathcal{B}$. The vector $\vec{\boldsymbol{\omega}}_{\mathcal{AB}}$ denotes the relative angular velocity of the coordinate system $\mathcal{B}$ w.r.t. the coordinate system $\mathcal{A}$. The skew symmetric matrix of a coordinate tuple $\boldsymbol{v} \in \mathbb{R}^3$ is denoted as $\boldsymbol{v}^\times \in \mathbb{R}^{3\times3}$ and has the property $\boldsymbol{v}^\times\boldsymbol{r} = \boldsymbol{v}\times\boldsymbol{r}\ \forall \boldsymbol{r} \in \mathbb{R}^3$, where $\times$ denotes the Euclidean cross-product. The term $\boldsymbol{v}^\times$ fulfills the following identities ($\boldsymbol{I} \in \mathbb{R}^{3\times3}$ is the identity matrix):

$$(\boldsymbol{v}^\times)^T = -\boldsymbol{v}^\times, \tag{5.1}$$

$$(\boldsymbol{v}^\times)^2 = \boldsymbol{v}\boldsymbol{v}^T - \boldsymbol{v}^T\boldsymbol{v}\boldsymbol{I}, \tag{5.2}$$

$$(\boldsymbol{C}(\Phi)\boldsymbol{v})^\times = \boldsymbol{C}(\Phi)\boldsymbol{v}^\times\boldsymbol{C}(\Phi)^T. \tag{5.3}$$

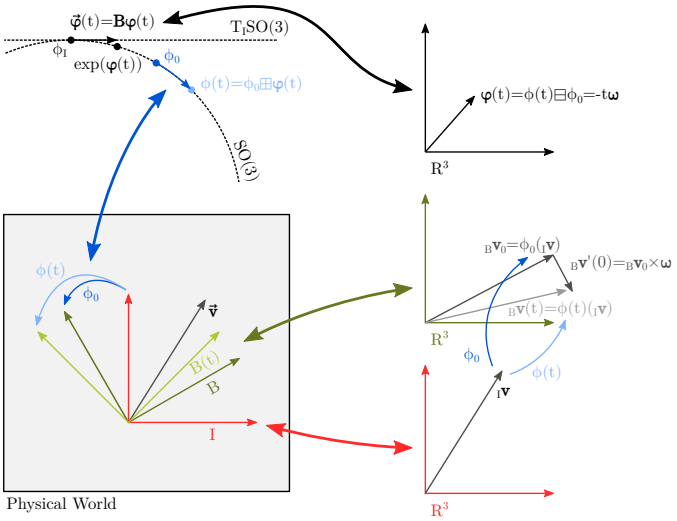

Figure 5.1: This figure depicts various quantities in a setup where a coordinate system $\mathcal{B}$ is rotated by a constant rotational velocity $\boldsymbol{\omega}$ w.r.t. an inertial coordinate system $\mathcal{I}$. Using coordinate systems, physical vectors can be represented through the corresponding coordinate tuples. Orientations between coordinate systems can be defined by the mapping they induce on the coordinate tuples. They are elements of $SO(3)$. Differences and derivatives of orientations can be represented in the tangential space $T_{\Phi_I} SO(3)$, which can be associated with $\mathbb{R}^3$ by means of a basis $\boldsymbol{B}$.

3 Theory

The following contemplations are independent of the choice of parametrization for 3D orientations. As will follow in the next definition, 3D orientations are first only thought of as mapping.

Given a 3D rigid body with attached body-fixed coordinate system $\mathcal{B}$, its orientation $\Phi_{\mathcal{BA}}$ w.r.t. a reference coordinate system $\mathcal{A}$ can be defined as the mapping of coordinates of any fixed vector $\vec{\boldsymbol{r}}$ from $\mathcal{A}$ to $\mathcal{B}$, that is,

$$_{\mathcal{B}}\boldsymbol{r} = \Phi_{\mathcal{BA}}(_{\mathcal{A}}\boldsymbol{r}). \tag{5.4}$$

Together with the concatenation operation, orientations form a Lie group known as the special orthogonal group $SO(3)$. The concatenation $\circ : SO(3) \times SO(3) \to SO(3)$ comes with the following (defining) identity:

$$(\Phi_{\mathcal{CB}} \circ \Phi_{\mathcal{BA}})(_{\mathcal{A}}\boldsymbol{r}) \triangleq \Phi_{\mathcal{CB}}(\Phi_{\mathcal{BA}}(_{\mathcal{A}}\boldsymbol{r})). \tag{5.5}$$

There also exists an identity element Φ_I and an inverse Φ^{-1} such that

$$\Phi_I \circ \Phi_{\mathcal{BA}} = \Phi_{\mathcal{BA}} \circ \Phi_I = \Phi_{\mathcal{BA}}, \tag{5.6}$$

$$\Phi_{\mathcal{BA}}^{-1} \circ \Phi_{\mathcal{BA}} = \Phi_{\mathcal{BA}} \circ \Phi_{\mathcal{BA}}^{-1} = \Phi_I. \tag{5.7}$$

The Lie group $SO(3)$ is not a vector space, has no addition operation, and consequently no subtraction either. This poses an issue if using orientations in filtering or optimization frameworks, which strongly rely on small differences and gradients (e.g. for linearization). Fortunately, since $SO(3)$ is a Lie group, there exists an exponential map $Exp : T_{\Phi_I}SO(3) \to SO(3)$ relating $SO(3)$ to its Lie algebra $T_{\Phi_I}SO(3)$. The later coincides with the tangent space at the identity element, which is isomorphic to $\mathbb{R}^3$. The exponential map is smooth and fulfills the following (uniquely) defining identities $\forall t, s \in \mathbb{R},\ \forall \vec{\boldsymbol{\varphi}} \in T_{\Phi_I}SO(3)$:

$$Exp((t+s)\vec{\boldsymbol{\varphi}}) = Exp(t\vec{\boldsymbol{\varphi}}) \circ Exp(s\vec{\boldsymbol{\varphi}}), \tag{5.8}$$

$$d/dt(Exp(t\vec{\boldsymbol{\varphi}}))|_{t=0} = \vec{\boldsymbol{\varphi}}. \tag{5.9}$$

Elements on $T_{\Phi_I}SO(3)$ are abstract vectors and are not very suitable for actual computations. By choosing a basis $\boldsymbol{B} = [\vec{\boldsymbol{\varphi}}_1, \vec{\boldsymbol{\varphi}}_2, \vec{\boldsymbol{\varphi}}_3]$ the map can be extended to $\mathbb{R}^3$. We define the exponential $\exp : \mathbb{R}^3 \to SO(3)$ of a coordinate tuple $\boldsymbol{\varphi} = (\varphi_1, \varphi_2, \varphi_3) \in \mathbb{R}^3$ by

$$\exp(\boldsymbol{\varphi}) := Exp(\vec{\boldsymbol{\varphi}}_1\varphi_1 + \vec{\boldsymbol{\varphi}}_2\varphi_2 + \vec{\boldsymbol{\varphi}}_3\varphi_3). \tag{5.10}$$

There is a certain degree of freedom in the selection of the basis $[\vec{\boldsymbol{\varphi}}_1, \vec{\boldsymbol{\varphi}}_2, \vec{\boldsymbol{\varphi}}_3]$. We choose the basis vectors $\vec{\boldsymbol{\varphi}}_i$ such that $\forall i \in \{1,2,3\}, \forall \boldsymbol{v} \in \mathbb{R}^3$:

$$d/dt(Exp(t\vec{\boldsymbol{\varphi}}_i)(\boldsymbol{v}))|_{t=0} = \boldsymbol{e}_i \times \boldsymbol{v} \tag{5.11}$$

where $\boldsymbol{e}_i \in \mathbb{R}^3$ are the standard basis vectors in $\mathbb{R}^3$. This makes $\exp(\cdot)$ a unique smooth mapping that fulfills $\forall t, s \in \mathbb{R},\ \forall \boldsymbol{\varphi}, \boldsymbol{v} \in \mathbb{R}^3$:

$$\exp((t+s)\boldsymbol{\varphi}) = \exp(t\boldsymbol{\varphi}) \circ \exp(s\boldsymbol{\varphi}) \tag{5.12}$$

$$d/dt(\exp(t\boldsymbol{\varphi})(\boldsymbol{v}))|_{t=0} = \boldsymbol{\varphi} \times \boldsymbol{v} \tag{5.13}$$

We will see later, that by using this definition of the exponential exp, its argument $\boldsymbol{\varphi}$ can be interpreted as the rotation vector associated with the relative orientation of two coordinate systems. There exists an open region around 0, the open ball with radius π $B_\pi(0) \subset \mathbb{R}^3$, such that the exponential is bijective and its image corresponds to all non-180°-orientations, $SO(3)^*$. Thus an inverse exists which is called the logarithm, $\log : SO(3)^* \to B_\pi(0)$.

With this we can construct boxplus and boxminus operations which adopt the function of addition and subtraction [59]:

$$\boxplus : SO(3) \times \mathbb{R}^3 \to SO(3), \tag{5.14}$$
$$\Phi, \boldsymbol{\varphi} \mapsto \exp(\boldsymbol{\varphi}) \circ \Phi,$$

$$\boxminus : SO(3) \times SO(3) \to \mathbb{R}^3, \tag{5.15}$$
$$\Phi_1, \Phi_2 \mapsto \log(\Phi_1 \circ \Phi_2^{-1}).$$

Similarly to regular addition and subtraction, both operators fulfill the following identities (axioms proposed by [59]):

$$\Phi \boxplus \mathbf{0} = \Phi, \tag{5.16}$$

$$(\Phi \boxplus \boldsymbol{\varphi}) \boxminus \Phi = \boldsymbol{\varphi}, \tag{5.17}$$

$$\Phi_1 \boxplus (\Phi_2 \boxminus \Phi_1) = \Phi_2. \tag{5.18}$$

This approach distinguishes between actual orientations, which are on $SO(3)$ (Lie group), and differences of orientations which lie on $\mathbb{R}^3$ (Lie algebra, see Figure 5.1). The above operators take care of appropriately transforming the elements into their respective spaces and allow a smooth embedding of rotational quantities in filtering and optimization frameworks.

The definition of differentials involving orientations can be adapted by replacing the regular plus and minus operators by the above boxplus and boxminus operators. For instance the differential of a mapping $f_1 : \mathbb{R} \to SO(3)$ can be defined as:

$$\frac{\partial}{\partial x} f_1(x) := \lim_{\epsilon \to 0} \frac{f_1(x+\epsilon) \boxminus f_1(x)}{\epsilon}. \tag{5.19}$$

The same can be done for the other case where we have a mapping $f_2 : SO(3) \to \mathbb{R}$:

$$\frac{\partial}{\partial \Phi} f_2(\Phi) := \lim_{\epsilon \to 0} \begin{bmatrix} \frac{f_2(\Phi \boxplus (\boldsymbol{e}_1 \epsilon)) - f_2(\Phi)}{\epsilon} \\ \frac{f_2(\Phi \boxplus (\boldsymbol{e}_2 \epsilon)) - f_2(\Phi)}{\epsilon} \\ \frac{f_2(\Phi \boxplus (\boldsymbol{e}_3 \epsilon)) - f_2(\Phi)}{\epsilon} \end{bmatrix}^T. \tag{5.20}$$

4 Implementation-Independent Identities

Some identities directly follow from the above considerations and are *independent* of the choice of the underlying orientation representation. By concatenating the exponential and the coordinate mapping we retrieve the well known Rodriguez' formula (see Appendix 7.1):

$$\begin{aligned} \boldsymbol{C}(\boldsymbol{\varphi}) &:= \boldsymbol{C}(\exp(\boldsymbol{\varphi})) \\ &= \boldsymbol{I} + \frac{\sin(\|\boldsymbol{\varphi}\|)\boldsymbol{\varphi}^\times}{\|\boldsymbol{\varphi}\|} + \frac{(1-\cos(\|\boldsymbol{\varphi}\|))\boldsymbol{\varphi}^{\times 2}}{\|\boldsymbol{\varphi}\|^2}, \end{aligned} \tag{5.21}$$

$$\boldsymbol{C}(\boldsymbol{\varphi}) \approx \boldsymbol{I} + \boldsymbol{\varphi}^\times, \quad (\|\boldsymbol{\varphi}\| \approx 0). \tag{5.22}$$

This shows that the argument of the exponential, $\boldsymbol{\varphi}$, can be interpreted as the coordinate tuple of the (*passive*) rotation vector associated with the relative orientation of two coordinate systems. Thus, if the corresponding coordinate systems are known we can write:

$$\Phi_{\mathcal{BA}} = \exp\left({}_{\mathcal{B}}\boldsymbol{\varphi}_{\mathcal{BA}}\right) = \exp\left({}_{\mathcal{A}}\boldsymbol{\varphi}_{\mathcal{BA}}\right). \tag{5.23}$$

We can also derive the following (adjoint related) identity (see Appendix 7.2):

$$\exp(\Phi(\boldsymbol{\varphi})) = \Phi \circ \exp(\boldsymbol{\varphi}) \circ \Phi^{-1}. \tag{5.24}$$

Useful identities can be derived for derivatives involving orientations (see Appendix 7):

$$\partial/\partial t\left(\Phi_{\mathcal{BA}}(t)\right) = -{}_{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{AB}}(t), \tag{5.25}$$
$$\partial/\partial \boldsymbol{r}\left(\Phi(\boldsymbol{r})\right) = \boldsymbol{C}(\Phi), \tag{5.26}$$
$$\partial/\partial \Phi\left(\Phi(\boldsymbol{r})\right) = -\left(\Phi(\boldsymbol{r})\right)^{\times}, \tag{5.27}$$
$$\partial/\partial \Phi\left(\Phi^{-1}\right) = -\boldsymbol{C}(\Phi)^{T}, \tag{5.28}$$
$$\partial/\partial \Phi_1\left(\Phi_1 \circ \Phi_2\right) = \boldsymbol{I}, \tag{5.29}$$
$$\partial/\partial \Phi_2\left(\Phi_1 \circ \Phi_2\right) = \boldsymbol{C}(\Phi_1), \tag{5.30}$$
$$\partial/\partial \boldsymbol{\varphi}\left(\exp(\boldsymbol{\varphi})\right) = \boldsymbol{\Gamma}(\boldsymbol{\varphi}), \tag{5.31}$$
$$\partial/\partial \Phi\left(\log(\Phi)\right) = \boldsymbol{\Gamma}^{-1}(\log(\Phi)). \tag{5.32}$$

The derivative of the exponential map is given by the Jacobian $\boldsymbol{\Gamma}(\boldsymbol{\varphi}) \in \mathbb{R}^{3\times 3}$ which has the following analytical expression:

$$\boldsymbol{\Gamma}(\boldsymbol{\varphi}) = \boldsymbol{I} + \frac{(1-\cos(\|\boldsymbol{\varphi}\|))\boldsymbol{\varphi}^{\times}}{\|\boldsymbol{\varphi}\|^2} + \frac{(\|\boldsymbol{\varphi}\| - \sin(\|\boldsymbol{\varphi}\|))\boldsymbol{\varphi}^{\times^2}}{\|\boldsymbol{\varphi}\|^3}, \tag{5.33}$$
$$\boldsymbol{\Gamma}(\boldsymbol{\varphi}) \approx \boldsymbol{I} + 1/2\boldsymbol{\varphi}^{\times}, \quad (\|\boldsymbol{\varphi}\| \approx 0). \tag{5.34}$$

5 Quaternion Implementation

The above discussion is completely decoupled from any actual orientation parameterization. It is valid whether Euler-angles, rotation matrices, quaternions, or other representations are employed. In the following we provide one possible *implementation* of 3D orientations along with the means to check its correctness. Here we propose the use of unit quaternions following the *Hamilton* convention [54] and we discuss the implementation of the different operations that are required. For more details on the differences between existing quaternion conventions we refer the reader to [125]. A unit quaternion is composed of a real part, $q_0 \in \mathbb{R}$, and an imaginary part, $\check{\boldsymbol{q}} \in \mathbb{R}^3$, which meet $q_0^2 + \|\check{\boldsymbol{q}}\|^2 = 1$. We denote this as $\Phi = (q_0, \check{\boldsymbol{q}})$.

5.1 Coordinates Mapping and Rotation Matrix

For arbitrary coordinate systems, $\mathcal{A}$ and $\mathcal{B}$, with relative orientation $\Phi_{\mathcal{BA}} = (q_0, \check{\boldsymbol{q}})$ the coordinates of a vector $\vec{\boldsymbol{r}}$ can be mapped as:

$$\Phi_{\mathcal{BA}}({}_{\mathcal{A}}\boldsymbol{r}) = (2q_0^2 - 1){}_{\mathcal{A}}\boldsymbol{r} + 2q_0\check{\boldsymbol{q}}^{\times}{}_{\mathcal{A}}\boldsymbol{r} + 2\check{\boldsymbol{q}}(\check{\boldsymbol{q}}^{T}{}_{\mathcal{A}}\boldsymbol{r}). \tag{5.35}$$

From this, we can directly derive the expression for the associated rotation matrix:

$$\boldsymbol{C}(\Phi_{\mathcal{BA}}) = (2q_0^2 - 1)\boldsymbol{I} + 2q_0\check{\boldsymbol{q}}^{\times} + 2\check{\boldsymbol{q}}\check{\boldsymbol{q}}^T. \tag{5.36}$$

5.2 Concatenation

The concatenation of two unit quaternions $\Phi_1 = (q_0, \check{\boldsymbol{q}})$ and $\Phi_2 = (p_0, \check{\boldsymbol{p}})$ is given by:

$$\Phi_1 \circ \Phi_2 = (q_0 p_0 - \check{\boldsymbol{q}}^T\check{\boldsymbol{p}}, q_0\check{\boldsymbol{p}} + p_0\check{\boldsymbol{q}} + \check{\boldsymbol{q}} \times \check{\boldsymbol{p}}). \tag{5.37}$$

5.3 Exponential and Logarithm

Given a $\boldsymbol{\varphi} \in \mathbb{R}^3$, the exponential map to a unit quaternion is given by:

$$\exp(\boldsymbol{\varphi}) = (q_0, \check{\boldsymbol{q}}) = \left(\cos(\|\boldsymbol{\varphi}\|/2), \sin(\|\boldsymbol{\varphi}\|/2)\frac{\boldsymbol{\varphi}}{\|\boldsymbol{\varphi}\|}\right) \tag{5.38}$$

$$\exp(\boldsymbol{\varphi}) \approx (1, \boldsymbol{\varphi}/2), \quad (\|\boldsymbol{\varphi}\| \approx 0). \tag{5.39}$$

The above small angle approximation is required to avoid numerical instabilities (typically for angles below 10^{-4} rad). The corresponding logarithm is given by:

$$\log(\Phi) = 2\,\mathrm{atan2}(\|\check{\boldsymbol{q}}\|, q_0)\frac{\check{\boldsymbol{q}}}{\|\check{\boldsymbol{q}}\|}, \tag{5.40}$$

$$\log(\Phi) \approx \mathrm{sign}(q_0)\,\check{\boldsymbol{q}}, \quad (\|\check{\boldsymbol{q}}\| \approx 0). \tag{5.41}$$

5.4 Consistency Tests

The consistency of the implementation can be tested through the following unit tests:

	$\Phi()$	$\boldsymbol{C}$	$\circ$	exp	log
$\boldsymbol{C}(\Phi)\boldsymbol{r} = \Phi(\boldsymbol{r})$	■	■			
$(\Phi_1 \circ \Phi_2)(\boldsymbol{r}) = \Phi_1(\Phi_2(\boldsymbol{r}))$	■		■		
$\boldsymbol{C}(\exp(\boldsymbol{\varphi})) = \boldsymbol{C}(\boldsymbol{\varphi})$		■		■	
$\Phi = \exp(\log(\Phi))$				■	■

On the right-hand side the involved operators are listed. The third test compares against Rodriguez' formula (eq. (5.21)). Theoretically, these tests should be carried out for all possible values of $\Phi, \Phi_1, \Phi_2 \in SO(3)$, $\boldsymbol{r}, \boldsymbol{\varphi} \in \mathbb{R}^3$. In practice, testing various samples, including very small angles, should be sufficient.

6 Simple Modeling Example

This section presents how to apply the above notation and convention to an actual system modeling task. We want to estimate the position, velocity (expressed in $\mathcal{B}$

to simplify the Jacobians), and orientation of a robot using an IMU and a generic position and orientation sensor (pose sensor). To avoid complicated modeling or specific knowledge about the motion model the IMU can be used to do a prediction of the state. This is very common in visual-inertial state estimation e.g. [14]. In the following, we first show how to use the IMU for predicting the state and then show the necessary steps to perform an update with the pose sensor.

6.1 Continuous Time Description

Let us assume we have an IMU driven dynamic system with inertial coordinate system $\mathcal{I}$ and IMU-fixed coordinate system $\mathcal{B}$ for which we wish to estimate the motion. Considering additive biases, ${}_{\mathcal{B}}\boldsymbol{b}_f$ and ${}_{\mathcal{B}}\boldsymbol{b}_\omega$, and using continuous-time white noise processes, ${}_{\mathcal{B}}\boldsymbol{n}_f$, ${}_{\mathcal{B}}\boldsymbol{n}_\omega$, ${}_{\mathcal{B}}\boldsymbol{n}_{bf}$, ${}_{\mathcal{B}}\boldsymbol{n}_{b\omega}$, we can model the IMU measurements, ${}_{\mathcal{B}}\tilde{\boldsymbol{f}}_{\mathcal{B}}$ and ${}_{\mathcal{B}}\tilde{\boldsymbol{\omega}}_{\mathcal{B}}$, as:

$$
\begin{aligned}
{}_{\mathcal{B}}\tilde{\boldsymbol{f}}_{\mathcal{B}} &= \Phi_{\mathcal{I}\mathcal{B}}^{-1}({}_{\mathcal{I}}\boldsymbol{a}_{\mathcal{B}} - {}_{\mathcal{I}}\boldsymbol{g}) + {}_{\mathcal{B}}\boldsymbol{b}_f + {}_{\mathcal{B}}\boldsymbol{n}_f, && (5.42)\\
{}_{\mathcal{B}}\tilde{\boldsymbol{\omega}}_{\mathcal{B}} &= {}_{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{I}\mathcal{B}} + {}_{\mathcal{B}}\boldsymbol{b}_\omega + {}_{\mathcal{B}}\boldsymbol{n}_\omega, && (5.43)\\
{}_{\mathcal{B}}\dot{\boldsymbol{b}}_f &= {}_{\mathcal{B}}\boldsymbol{n}_{bf}, && (5.44)\\
{}_{\mathcal{B}}\dot{\boldsymbol{b}}_\omega &= {}_{\mathcal{B}}\boldsymbol{n}_{b\omega}, && (5.45)
\end{aligned}
$$

where ${}_{\mathcal{I}}\boldsymbol{g}$ is the gravity vector expressed in the inertial frame. We add the IMU biases to the state $\boldsymbol{x}$. This gives the full state by

$$
\boldsymbol{x} = \left({}_{\mathcal{I}}\boldsymbol{r}_{\mathcal{I}\mathcal{B}},\, {}_{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}},\, \Phi_{\mathcal{I}\mathcal{B}},\, {}_{\mathcal{B}}\boldsymbol{b}_f,\, {}_{\mathcal{B}}\boldsymbol{b}_\omega\right). \tag{5.46}
$$

The resulting continuous-time equations of motion can be written as:

$$
\begin{aligned}
{}_{\mathcal{I}}\dot{\boldsymbol{r}}_{\mathcal{I}\mathcal{B}} &= \Phi_{\mathcal{I}\mathcal{B}}({}_{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}} + {}_{\mathcal{B}}\boldsymbol{n}_v), && (5.47)\\
{}_{\mathcal{B}}\dot{\boldsymbol{v}}_{\mathcal{B}} &= {}^{d}/_{dt}\left(\Phi_{\mathcal{I}\mathcal{B}}^{-1}({}_{\mathcal{I}}\boldsymbol{v}_{\mathcal{B}})\right)\\
&= \Phi_{\mathcal{I}\mathcal{B}}^{-1}({}_{\mathcal{I}}\dot{\boldsymbol{v}}_{\mathcal{B}}) - \left(\Phi_{\mathcal{I}\mathcal{B}}^{-1}({}_{\mathcal{I}}\boldsymbol{v}_{\mathcal{B}})\right)^{\times}\boldsymbol{C}(\Phi_{\mathcal{I}\mathcal{B}})^{T}{}_{\mathcal{I}}\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}\\
&= \Phi_{\mathcal{I}\mathcal{B}}^{-1}({}_{\mathcal{I}}\boldsymbol{a}_{\mathcal{B}}) - {}_{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}}^{\times}{}_{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}\\
&= \Phi_{\mathcal{I}\mathcal{B}}^{-1}({}_{\mathcal{I}}\boldsymbol{g}) + {}_{\mathcal{B}}\boldsymbol{f}_{\mathcal{I}\mathcal{B}} - {}_{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{I}\mathcal{B}}^{\times}{}_{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}}, && (5.48)\\
\dot{\Phi}_{\mathcal{I}\mathcal{B}} &= -{}_{\mathcal{I}}\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}} = \Phi_{\mathcal{I}\mathcal{B}}({}_{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{I}\mathcal{B}}), && (5.49)\\
{}_{\mathcal{B}}\dot{\boldsymbol{b}}_f &= {}_{\mathcal{B}}\boldsymbol{n}_{bf}, && (5.50)\\
{}_{\mathcal{B}}\dot{\boldsymbol{b}}_\omega &= {}_{\mathcal{B}}\boldsymbol{n}_{b\omega}, && (5.51)
\end{aligned}
$$

with the bias and noise corrected proper acceleration and angular velocity measurements

$$
\begin{aligned}
{}_{\mathcal{B}}\boldsymbol{f}_{\mathcal{I}\mathcal{B}} &= {}_{\mathcal{B}}\tilde{\boldsymbol{f}}_{\mathcal{B}} - {}_{\mathcal{B}}\boldsymbol{b}_f - {}_{\mathcal{B}}\boldsymbol{n}_f, && (5.52)\\
{}_{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{I}\mathcal{B}} &= {}_{\mathcal{B}}\tilde{\boldsymbol{\omega}}_{\mathcal{B}} - {}_{\mathcal{B}}\boldsymbol{b}_\omega - {}_{\mathcal{B}}\boldsymbol{n}_\omega. && (5.53)
\end{aligned}
$$

To derive (5.48) we used the product rule, followed by the chain rule and the identities (5.25),(5.27),(5.28).

6.2 Euler-Forward Discretization

One of the simplest and most commonly used discretization methods is Euler-Forward discretization. Other discretization schemes can of course also be employed. For a time increment Δt, Euler-Foward discretization of the above formulation yields (the next state is denoted by a bar, discretized noise by a hat):

$$
\begin{aligned}
{}_{\mathcal{I}}\bar{\boldsymbol{r}}_{\mathcal{IB}} &= {}_{\mathcal{I}}\boldsymbol{r}_{\mathcal{IB}} + \Delta t\ \Phi_{\mathcal{IB}}({}_{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}} + {}_{\mathcal{B}}\hat{\boldsymbol{n}}_v), && (5.54)\\
{}_{\mathcal{B}}\bar{\boldsymbol{v}}_{\mathcal{B}} &= {}_{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}} + \Delta t\Big(\Phi^{-1}_{\mathcal{IB}}({}_{\mathcal{I}}\boldsymbol{g}) + \boldsymbol{f} - \boldsymbol{\omega}^{\times}\,{}_{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}}\Big) && (5.55)\\
\bar{\Phi}_{\mathcal{IB}} &= \Phi_{\mathcal{IB}} \boxplus (\Delta t \Phi_{\mathcal{IB}}(\boldsymbol{\omega}))\\
&= \exp(\Phi_{\mathcal{IB}}(\Delta t\boldsymbol{\omega})) \circ \Phi_{\mathcal{IB}}\\
&= \Phi_{\mathcal{IB}} \circ \exp(\Delta t\boldsymbol{\omega}) \circ \Phi^{-1}_{\mathcal{IB}} \circ \Phi_{\mathcal{IB}}\\
&= \Phi_{\mathcal{IB}} \circ \exp(\Delta t\boldsymbol{\omega}), && (5.56)\\
{}_{\mathcal{B}}\bar{\boldsymbol{b}}_f &= {}_{\mathcal{B}}\boldsymbol{b}_f + \ \Delta t_{\mathcal{B}}\hat{\boldsymbol{n}}_{bf}, && (5.57)\\
{}_{\mathcal{B}}\bar{\boldsymbol{b}}_\omega &= {}_{\mathcal{B}}\boldsymbol{b}_\omega + \ \Delta t_{\mathcal{B}}\hat{\boldsymbol{n}}_{b\omega}, && (5.58)
\end{aligned}
$$

with the discretized IMU measurements (bias and noise corrected) given by

$$
\begin{aligned}
\boldsymbol{f} &= {}_{\mathcal{B}}\tilde{\boldsymbol{f}}_{\mathcal{B}} - {}_{\mathcal{B}}\boldsymbol{b}_f - {}_{\mathcal{B}}\hat{\boldsymbol{n}}_f, && (5.59)\\
\boldsymbol{\omega} &= {}_{\mathcal{B}}\tilde{\boldsymbol{\omega}}_{\mathcal{B}} - {}_{\mathcal{B}}\boldsymbol{b}_\omega - {}_{\mathcal{B}}\hat{\boldsymbol{n}}_\omega. && (5.60)
\end{aligned}
$$

The noise is discretized such that, if $\boldsymbol{R}_i$ is the noise density of the white noise process $\boldsymbol{n}_i$, then the discrete Gaussian noise $\hat{\boldsymbol{n}}_i$ is distributed with $\mathcal{N}(0, \boldsymbol{R}_i/\Delta t)$.

6.3 Differentiation

Using the identities (5.25)-(5.32) and applying the chain rule, the following Jacobians of the discrete process model can be derived ($\boldsymbol{F}$ is w.r.t. the state, $\boldsymbol{G}$ is w.r.t. the process noise):

$$
\boldsymbol{F} = \begin{bmatrix}
\boldsymbol{I} & \Delta t\,\boldsymbol{C}(\Phi_{\mathcal{IB}}) & -\Delta t\,\Phi_{\mathcal{IB}}({}_{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}})^{\times} & 0 & 0\\
0 & \boldsymbol{I} - \Delta t\boldsymbol{\omega}^{\times} & \Delta t\,\boldsymbol{C}(\Phi_{\mathcal{IB}})^T({}_{\mathcal{I}}\boldsymbol{g})^{\times} & -\Delta t\,\boldsymbol{I} & -\Delta t\,{}_{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}}^{\times}\\
0 & 0 & \boldsymbol{I} & 0 & -\Delta t\boldsymbol{C}(\Phi_{\mathcal{IB}})\boldsymbol{\Gamma}(\Delta t\boldsymbol{\omega})\\
0 & 0 & 0 & \boldsymbol{I} & 0\\
0 & 0 & 0 & 0 & \boldsymbol{I}
\end{bmatrix},
$$

$$
\boldsymbol{G} = \begin{bmatrix}
\Delta t\,\boldsymbol{C}(\Phi_{\mathcal{IB}}) & 0 & 0 & 0 & 0\\
0 & -\Delta t\boldsymbol{I} & -\Delta t({}_{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}})^{\times} & 0 & 0\\
0 & 0 & -\Delta t\boldsymbol{C}(\Phi_{\mathcal{IB}})\boldsymbol{\Gamma}(\Delta t\boldsymbol{\omega}) & 0 & 0\\
0 & 0 & 0 & \Delta t\boldsymbol{I} & 0\\
0 & 0 & 0 & 0 & \Delta t\boldsymbol{I}
\end{bmatrix},
$$

6.4 Measurement

For simplicity we assume a GPS position measurement ${}_{\mathcal{I}}\tilde{\boldsymbol{r}}_{\mathcal{IB}}$ and an orientation measurement $\tilde{\Phi}_{\mathcal{IB}}$. The measurement equations are given by

$$
\begin{aligned}
{}_{\mathcal{I}}\tilde{\boldsymbol{r}}_{\mathcal{IB}} &= {}_{\mathcal{I}}\boldsymbol{r}_{\mathcal{IB}} + {}_{\mathcal{I}}\hat{\boldsymbol{n}}_{p}, && (5.61)\\
\tilde{\Phi}_{\mathcal{IB}} &= \Phi_{\mathcal{IB}} \boxplus {}_{\mathcal{I}}\hat{\boldsymbol{n}}_{\Phi} && (5.62)\\
&= \exp({}_{\mathcal{I}}\hat{\boldsymbol{n}}_{\Phi}) \circ \Phi_{\mathcal{IB}}, && (5.63)
\end{aligned}
$$

with the discrete Gaussian measurement noise vectors ${}_{\mathcal{I}}\hat{\boldsymbol{n}}_{p}$ and ${}_{\mathcal{I}}\hat{\boldsymbol{n}}_{\Phi}$.

Using the identities (5.29),(5.30),(5.31) and because the expectation of the orientation measurement noise is zero, the following Jacobians can be derived ($\boldsymbol{H}$ is w.r.t. the state, $\boldsymbol{J}$ is w.r.t. the update noise):

$$
\boldsymbol{H} = \begin{bmatrix} \boldsymbol{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & \boldsymbol{I} & 0 & 0 \end{bmatrix},
$$

$$
\boldsymbol{J} = \begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix},
$$

6.5 Hints for the EKF Implementation

Now that we have derived all the required parts, the well known EKF equations can be used to estimate the state. The only difference to the standard EKF is that we need to use the $\boxminus$ operator for the innovation residual and the $\boxplus$ operator for updating the state estimate instead of normal addition and subtraction.

7 Conclusion

This document derived and summarized the main identities related to 3D orientations in robotics and other engineering fields. In particular it discussed a more abstract but convention-less notion of 3D orientations, the boxplus and boxminus operators, as well as the concept of differentials. Various differentials involving 3D orientations are derived, which can be used to compute the Jacobians of more complex models by applying the chain rule. A simple modeling example shows how to apply the introduced concepts.

Derivatives Involving Orientations

Time Derivative of Orientation

Here we need the kinematic concept of angular velocities. We assume the existence of an inertial observer $\mathcal{I}$ which observes the motion, over a duration ϵ, of a moving coordinate system $\mathcal{B}(t)$. We use the following definition of angular velocities (the negative sign is required so that the angular velocity corresponds to the *active* rotation which is measured by typical IMU devices):

$$
{}_{\mathcal{B}(t)}\boldsymbol{\omega}_{\mathcal{IB}(t)} := -\lim_{\epsilon \to 0} \frac{{}_{\mathcal{B}(t)}\boldsymbol{\varphi}_{\mathcal{B}(t+\epsilon)\mathcal{B}(t)}}{\epsilon} \tag{5.64}
$$

Additionally we require the limit (based on the limits (5.39),(5.41)):

$$\lim_{\epsilon\to 0}\frac{\log(\exp(\epsilon\boldsymbol{\varphi}_1)\circ\exp(\epsilon\boldsymbol{\varphi}_2))}{\epsilon}=\boldsymbol{\varphi}_1+\boldsymbol{\varphi}_2. \tag{5.65}$$

With this we can derive the derivative of an orientation $\Phi_{\mathcal{B}(t)\mathcal{A}(t)}$ w.r.t. time t (used identities: (5.19),(5.15),(5.23),(5.24),(5.64),(5.65)):

$$\begin{aligned}
\frac{\partial}{\partial t}\Phi_{\mathcal{B}(t)\mathcal{A}(t)} &= \lim_{\epsilon\to 0}\frac{\Phi_{\mathcal{B}(t+\epsilon)\mathcal{A}(t+\epsilon)}\boxminus\Phi_{\mathcal{B}(t)\mathcal{A}(t)}}{\epsilon}\\
&= \lim_{\epsilon\to 0}\frac{1}{\epsilon}\Big((\Phi_{\mathcal{B}(t+\epsilon)\mathcal{B}(t)}\circ\Phi_{\mathcal{B}(t)\mathcal{A}(t)}\circ\Phi_{\mathcal{A}(t)\mathcal{A}(t+\epsilon)})\\
&\qquad\boxminus\Phi_{\mathcal{B}(t)\mathcal{A}(t)}\Big)\\
&= \lim_{\epsilon\to 0}\frac{1}{\epsilon}\Big(\log(\Phi_{\mathcal{B}(t+\epsilon)\mathcal{B}(t)}\circ\Phi_{\mathcal{B}(t)\mathcal{A}(t)}\\
&\qquad\circ\Phi_{\mathcal{A}(t)\mathcal{A}(t+\epsilon)}\circ\Phi^{-1}_{\mathcal{B}(t)\mathcal{A}(t)})\Big)\\
&= \lim_{\epsilon\to 0}\frac{1}{\epsilon}\Big(\log(\exp({}_{\mathcal{B}(t)}\boldsymbol{\varphi}_{\mathcal{B}(t+\epsilon)\mathcal{B}(t)})\circ\Phi_{\mathcal{B}(t)\mathcal{A}(t)}\\
&\qquad\circ\exp({}_{\mathcal{A}(t)}\boldsymbol{\varphi}_{\mathcal{A}(t)\mathcal{A}(t+\epsilon)})\circ\Phi^{-1}_{\mathcal{B}(t)\mathcal{A}(t)})\Big)\\
&= \lim_{\epsilon\to 0}\frac{1}{\epsilon}\Big(\log(\exp({}_{\mathcal{B}(t)}\boldsymbol{\varphi}_{\mathcal{B}(t+\epsilon)\mathcal{B}(t)})\\
&\qquad\circ\exp({}_{\mathcal{B}(t)}\boldsymbol{\varphi}_{\mathcal{A}(t)\mathcal{A}(t+\epsilon)}))\Big)\\
&= \lim_{\epsilon\to 0}\frac{1}{\epsilon}\Big(\log(\exp(-\epsilon\,{}_{\mathcal{B}(t)}\boldsymbol{\omega}_{\mathcal{I}\mathcal{B}(t)})\\
&\qquad\circ\exp(\epsilon\,{}_{\mathcal{B}(t)}\boldsymbol{\omega}_{\mathcal{I}\mathcal{A}(t)}))\Big)\\
&= -\,{}_{\mathcal{B}(t)}\boldsymbol{\omega}_{\mathcal{I}\mathcal{B}(t)}+{}_{\mathcal{B}(t)}\boldsymbol{\omega}_{\mathcal{I}\mathcal{A}(t)}\\
&= -\,{}_{\mathcal{B}(t)}\boldsymbol{\omega}_{\mathcal{A}(t)\mathcal{B}(t)}
\end{aligned}\tag{5.66}$$

Derivative of Inverse

Here we derive the derivative of the inverse of an orientation (used identities: (5.19), (5.20), (5.15), (5.14), (5.24)):

$$\begin{aligned}
\left[\frac{\partial}{\partial\Phi}\Phi^{-1}\right]_i &= \lim_{\epsilon\to 0}\frac{(\Phi\boxplus\boldsymbol{e}_i\epsilon)^{-1}\boxminus\Phi^{-1}}{\epsilon}\\
&= \lim_{\epsilon\to 0}\frac{\log(\Phi^{-1}\circ\exp(-\boldsymbol{e}_i\epsilon)\circ\Phi)}{\epsilon}\\
&= \lim_{\epsilon\to 0}\frac{\log(\exp(-\Phi^{-1}(\boldsymbol{e}_i)\epsilon)}{\epsilon}\\
&= -\,\Phi^{-1}(\boldsymbol{e}_i)=-\boldsymbol{C}(\Phi)^T\boldsymbol{e}_i.
\end{aligned}$$

$$\frac{\partial}{\partial \Phi}\Phi^{-1} = -\boldsymbol{C}(\Phi)^T. \tag{5.67}$$

Derivative of Coordinate Map

The map of an orientation applied to a coordinate tuple can be differentiated w.r.t. the orientation itself. This yields (used identities: (5.20), (5.14), (5.5), (5.22)):

$$\begin{aligned}
\left[\frac{\partial}{\partial \Phi}\Phi(\boldsymbol{r})\right]_i &= \lim_{\epsilon\to 0}\frac{(\Phi \boxplus \boldsymbol{e}_i\epsilon)(\boldsymbol{r}) - \Phi(\boldsymbol{r})}{\epsilon} \\
&= \lim_{\epsilon\to 0}\frac{\boldsymbol{C}(\boldsymbol{e}_i\epsilon)\boldsymbol{C}(\Phi)\boldsymbol{r} - \boldsymbol{C}(\Phi)\boldsymbol{r}}{\epsilon} \\
&= \lim_{\epsilon\to 0}\frac{(\boldsymbol{I} + \boldsymbol{e}_i^{\times}\epsilon)\boldsymbol{C}(\Phi)\boldsymbol{r} - \boldsymbol{C}(\Phi)\boldsymbol{r}}{\epsilon} \\
&= \lim_{\epsilon\to 0}\frac{\boldsymbol{e}_i^{\times}\epsilon\boldsymbol{C}(\Phi)\boldsymbol{r}}{\epsilon} \\
&= -(\boldsymbol{C}(\Phi)\boldsymbol{r})^{\times}\boldsymbol{e}_i. \\
\frac{\partial}{\partial \Phi}\Phi(\boldsymbol{r}) &= -(\boldsymbol{C}(\Phi)\boldsymbol{r})^{\times}.
\end{aligned} \tag{5.68}$$

Concatenation - Left

The concatenation of two orientations can be differentiated w.r.t. the involved orientations. We first derive the derivative w.r.t. the left orientation (used identities: (5.19),(5.20),(5.15),(5.14)):

$$\begin{aligned}
\left[\frac{\partial}{\partial \Phi_1}\Phi_1 \circ \Phi_2\right]_i &= \lim_{\epsilon\to 0}\frac{((\Phi_1 \boxplus \boldsymbol{e}_i\epsilon)\circ\Phi_2) \boxminus (\Phi_1 \circ \Phi_2)}{\epsilon} \\
&= \lim_{\epsilon\to 0}\frac{\log(\exp(\boldsymbol{e}_i\epsilon)\circ\Phi_1\circ\Phi_2\circ\Phi_2^{-1}\circ\Phi_1^{-1})}{\epsilon} \\
&= \boldsymbol{e}_i. \\
\frac{\partial}{\partial \Phi_1}\Phi_1 \circ \Phi_2 &= \boldsymbol{I}.
\end{aligned} \tag{5.69}$$

Concatenation - Right

The derivative of the concatenation w.r.t. the right orientation yields (used identities: (5.19),(5.20),(5.15),(5.14),(5.24)):

$$\begin{aligned}
\left[\frac{\partial}{\partial \Phi_2}\Phi_1 \circ \Phi_2\right]_i &= \lim_{\epsilon\to 0}\frac{(\Phi_1\circ(\Phi_2 \boxplus \boldsymbol{e}_i\epsilon)) \boxminus (\Phi_1 \circ \Phi_2)}{\epsilon} \\
&= \lim_{\epsilon\to 0}\frac{\log(\Phi_1\circ\exp(\boldsymbol{e}_i\epsilon)\circ\Phi_2\circ\Phi_2^{-1}\circ\Phi_1^{-1})}{\epsilon} \\
&= \lim_{\epsilon\to 0}\frac{\log(\exp(\Phi_1(\boldsymbol{e}_i)\epsilon))}{\epsilon}
\end{aligned}$$

$$= \Phi_1(\boldsymbol{e}_i) = \boldsymbol{C}(\Phi_1)\boldsymbol{e}_i.$$

$$\frac{\partial}{\partial \Phi_2}\Phi_1 \circ \Phi_2 = \boldsymbol{C}(\Phi_1). \tag{5.70}$$

Exponential Derivative

Define:

$$\boldsymbol{\Gamma}(\boldsymbol{\varphi}) := \partial/\partial\boldsymbol{\varphi}\left(\exp(\boldsymbol{\varphi})\right). \tag{5.71}$$

Differentiate the adjoint related identity using the chain rule and product rule (identities (5.71),(5.27) for left side, identities (5.29),(5.30),(5.28) for right side):

$$\partial/\partial\Phi\Big[\exp(\Phi(\boldsymbol{\varphi})) = \Phi \circ \exp(\boldsymbol{\varphi}) \circ \Phi^{-1}\Big], \tag{5.72}$$

$$-\boldsymbol{\Gamma}(\Phi(\boldsymbol{\varphi}))\Phi(\boldsymbol{\varphi})^{\times} = \boldsymbol{I} - \boldsymbol{C}(\Phi)\boldsymbol{C}(\boldsymbol{\varphi})\boldsymbol{C}(\Phi)^T. \tag{5.73}$$

Set Φ to identity:

$$\boldsymbol{\Gamma}(\boldsymbol{\varphi})\boldsymbol{\varphi}^{\times} = \boldsymbol{C}(\boldsymbol{\varphi}) - \boldsymbol{I}. \tag{5.74}$$

Now consider the map $f(x) = \exp(x\boldsymbol{\varphi})$ for some arbitrary $\boldsymbol{\varphi} \in \mathbb{R}^3$. The chain rule yields $f'(x) = \boldsymbol{\Gamma}(x\boldsymbol{\varphi})\boldsymbol{\varphi}$. Alternatively, it can be differentiated using the limit (5.19) (used identities: (5.8),(5.15)):

$$\begin{aligned} f'(x) &= \lim_{\epsilon\to 0} \frac{\exp((x+\epsilon)\boldsymbol{\varphi}) \boxminus \exp(x\boldsymbol{\varphi})}{\epsilon} \\ &= \lim_{\epsilon\to 0} \frac{\log(\exp(\epsilon\boldsymbol{\varphi}) \circ \exp(x\boldsymbol{\varphi}) \circ \exp(x\boldsymbol{\varphi})^{-1})}{\epsilon} \\ &= \boldsymbol{\varphi}. \end{aligned} \tag{5.75}$$

Compare both derivatives at $x = 1$:

$$\boldsymbol{\Gamma}(\boldsymbol{\varphi})\boldsymbol{\varphi} = \boldsymbol{\varphi}. \tag{5.76}$$

This can be combined with eq. (5.74) in order to obtain the following matrix equation:

$$\boldsymbol{\Gamma}(\boldsymbol{\varphi})\begin{bmatrix}\boldsymbol{\varphi}^{\times} & \boldsymbol{\varphi}\end{bmatrix} = \begin{bmatrix}\boldsymbol{C}(\boldsymbol{\varphi}) - \boldsymbol{I} & \boldsymbol{\varphi}\end{bmatrix}. \tag{5.77}$$

Right multiply with $\begin{bmatrix}\boldsymbol{\varphi}^{\times} & \boldsymbol{\varphi}\end{bmatrix}^T$ and simplify:

$$\boldsymbol{\Gamma}(\boldsymbol{\varphi})(-{\boldsymbol{\varphi}^{\times}}^2 + \boldsymbol{\varphi}\boldsymbol{\varphi}^T) = (\boldsymbol{I} - \boldsymbol{C}(\boldsymbol{\varphi}))\boldsymbol{\varphi}^{\times} + \boldsymbol{\varphi}\boldsymbol{\varphi}^T, \tag{5.78}$$

$$\boldsymbol{\Gamma}(\boldsymbol{\varphi})\|\boldsymbol{\varphi}\|^2 = (\boldsymbol{I} - \boldsymbol{C}(\boldsymbol{\varphi}))\boldsymbol{\varphi}^{\times} + \boldsymbol{\varphi}\boldsymbol{\varphi}^T, \tag{5.79}$$

$$\boldsymbol{\Gamma}(\boldsymbol{\varphi}) = \frac{(\boldsymbol{I} - \boldsymbol{C}(\boldsymbol{\varphi}))\boldsymbol{\varphi}^{\times} + \boldsymbol{\varphi}\boldsymbol{\varphi}^T}{\|\boldsymbol{\varphi}\|^2}. \tag{5.80}$$

If substituting $\boldsymbol{C}(\boldsymbol{\varphi})$ we obtain eq. (5.33).

Other Proofs

7.1 Rodriguez' Formula

From eqs. (5.5), (5.12) and (5.13) we obtain the following properties for $\boldsymbol{C}(\boldsymbol{\varphi}) = \boldsymbol{C}(\exp(\boldsymbol{\varphi}))$, $\forall t \in \mathbb{R}$, $\boldsymbol{\varphi}, \boldsymbol{v} \in \mathbb{R}^3$:

$$\boldsymbol{C}((t+s)\boldsymbol{\varphi}) = \boldsymbol{C}(t\boldsymbol{\varphi})\boldsymbol{C}(s\boldsymbol{\varphi}) \tag{5.81}$$

$$d/dt\left(\boldsymbol{C}(t\boldsymbol{\varphi})(\boldsymbol{v})\right)|_{t=0} = \boldsymbol{\varphi} \times \boldsymbol{v} \tag{5.82}$$

For a given $\boldsymbol{\varphi}$ we define the curve $\boldsymbol{C}_{\boldsymbol{\varphi}}(t) := \boldsymbol{C}(t\boldsymbol{\varphi})$. Using a change of coordinate $t = s + r$, we can extend the range of the differential identity $\forall t \in \mathbb{R}$, $\boldsymbol{v} \in \mathbb{R}^3$:

$$d/dt\left(\boldsymbol{C}(t\boldsymbol{\varphi})\boldsymbol{v}\right) = d/ds\left(\boldsymbol{C}(s\boldsymbol{\varphi})\boldsymbol{C}(r\boldsymbol{\varphi})\boldsymbol{v}\right)|_{s=0,r=t} \tag{5.83}$$

$$= \boldsymbol{\varphi}^{\times}\boldsymbol{C}(t\boldsymbol{\varphi})\boldsymbol{v} \tag{5.84}$$

Thus, we obtain the following matrix differential equation:

$$d/dt\left(\boldsymbol{C}_{\boldsymbol{\varphi}}(t)\right) = \boldsymbol{\varphi}^{\times}\boldsymbol{C}_{\boldsymbol{\varphi}}(t), \tag{5.85}$$

which has the matrix exponential solution

$$\boldsymbol{C}_{\boldsymbol{\varphi}}(t) = e^{t\boldsymbol{\varphi}^{\times}}. \tag{5.86}$$

Since this is valid for arbitrary $\boldsymbol{\varphi}$, we obtain:

$$\boldsymbol{C}(\boldsymbol{\varphi}) = e^{\boldsymbol{\varphi}^{\times}}, \tag{5.87}$$

which can be shown to be the same as eq. (5.21) using series expansions.

7.2 Concatenation and Exponential – Adjoint Related

We want to prove the following identity:

$$\exp(\Phi(\boldsymbol{\varphi})) = \Phi \circ \exp(\boldsymbol{\varphi}) \circ \Phi^{-1}. \tag{5.88}$$

Since we know that exp is unique it is sufficient to show that the right hand side is indeed the exponential of $\Phi(\boldsymbol{\varphi})$ and thus check the defining properties. First we verify eq. (5.12):

$$\exp((t+s)\Phi(\boldsymbol{\varphi})) = \Phi \circ \exp((t+s)\boldsymbol{\varphi}) \circ \Phi^{-1} \tag{5.89}$$

$$= \Phi \circ \exp(t\boldsymbol{\varphi}) \circ \exp(s\boldsymbol{\varphi}) \circ \Phi^{-1} \tag{5.90}$$

$$= \Phi \circ \exp(t\boldsymbol{\varphi}) \circ \Phi^{-1} \circ \Phi \circ \exp(s\boldsymbol{\varphi}) \circ \Phi^{-1} \tag{5.91}$$

$$= \exp(t\Phi(\boldsymbol{\varphi})) \circ \exp(s\Phi(\boldsymbol{\varphi})). \tag{5.92}$$

Equation (5.13) poses a requirement on the derivative which can also be verified:

$$
\begin{aligned}
{}^{d}/_{dt}\left(\exp(t\Phi(\boldsymbol{\varphi}))(\boldsymbol{v})\right)|_{t=0} &= {}^{d}/_{dt}\left(\boldsymbol{C}(\Phi)\left(\exp(t\boldsymbol{\varphi})(\boldsymbol{C}(\Phi)^{T}\boldsymbol{v})\right)\right)|_{t=0} && (5.93)\\
&= \boldsymbol{C}(\Phi)\boldsymbol{\varphi}^{\times}\boldsymbol{C}(\Phi)^{T}\boldsymbol{v} && (5.94)\\
&= (\boldsymbol{C}(\Phi)\boldsymbol{\varphi})^{\times}\boldsymbol{v} = \Phi(\boldsymbol{\varphi}) \times \boldsymbol{v}. && (5.95)
\end{aligned}
$$

Since $\Phi \circ \exp(\boldsymbol{\varphi}) \circ \Phi^{-1}$ fulfills both uniquely defining properties of the exponential it is indeed equivalent to $\exp(\Phi(\boldsymbol{\varphi}))$.

Paper II

Technical Implementations of the Sense of Balance

Michael Bloesch, Marco Hutter

Abstract

Control algorithms for legged robots rely on accurate and fail-safe egomotion estimation in order to keep balance and perform desired tasks. To this end, the robot must integrate the measurements from different sensor modalities into a single consistent state estimation. In particular, the estimation process must provide estimates of the gravity direction and the local velocities of the robot since those quantities are essential for stabilizing the system and to counteract external disturbances. In comparison to other types of robots, legged robots interact through intermittent contacts with the surrounding. This provides the system with an additional source of information which can be leveraged in order to improve the state estimation. Since there is no one-size-fits-all solution, the following chapter will provide an insight into the different concepts and algorithms by discussing state-of-the-art approaches and examples. This should enable the reader to design a tailored state estimation solution to his or her specific robot and environment.

Published in:
Humanoid Robotics: a *Reference*, Springer 2017

1 Introduction

The capability of estimating a robot's posture and ego-motion with respect to its surrounding is an essential part when operating humanoid robots. Especially for dynamically unstable robots, where the robot is required to constantly counteract the effects of gravity and potential disturbances, it is of highest importance to estimate the gravity direction as well as the ego-motion of the robot. In particular, it is also essential to estimate linear and rotational velocities since these quantities are very well suited to detect the occurrence of a disturbance and its effect on the robot (e.g. a push to the left). The superimposed feedback loop imposes specifications on the accuracy and bandwidth of the estimation process, whereas estimation failures can quickly lead to damaging of the robot and its surrounding.

Looking at human posture estimation, one can observe that they constantly integrate information they obtain from different sensor modalities. This includes proprioceptive information from joint position and stress sensors, linear and rotational acceleration measurements from the vestibular system, as well as visual information from the eyes. Although it is still unclear how this information gets processed in detail, different researcher could show that the single modalities get weighted and combined in the central nervous system [109]. This results in a reliable sense of balance, where the weighting can be adapted to the actual task and environment. E.g. if a healthy human is walking on soft or unstable terrain he will automatically down-weight the proprioceptive information and rely more on his visual and vestibular senses. Still, in the presence of strong perturbations the system can fail leading to loss of balance and nausea as can be observed in the case of seasick individuals.

In the broader robotics community various sensor modalities have been employed for ego-motion estimation, many of which have a human counterpart (see fig. 6.1). For instance, the widely used accelerometers measure linear accelerations, which are also measured by the human vestibular system. The human counterpart of a robot's joint encoders and stress sensors can be found in the human muscle and joint proprioceptive sensors. Furthermore, it is not difficult to make the connection between cameras and human eyes. But a robot's range of sensor modalities goes beyond this and for instance includes range sensing (also referred to as Lidar) which does not have a human counterpart.

Since a lot of concept described in this chapter do not depend on the number of legs, the discussion on the sense of balance will be kept in a broader context of legged robots. All legged robots have in common that they interact with their surrounding by the mean of intermittent ground contact. Those intermittent ground contact will be central to this chapter since they represent the main difference compared to other types of robots like flying or wheeled robots. An important aspect is given by the type of foot model that can be employed. This influences the amount of information that can be retrieved from a contact with the ground.

This chapter is structured as follows. The first two sections introduce modeling of legged systems and provide a brief introduction on sensor fusion algorithms. After that, different state-of-the-art methods for doing state estimation with legged robots will be discussed in order to provide the reader with an overview of the available tools and concepts. This will include methodologies ranging from pure kinematic odometry to the inclusion of dynamic information or fusion with other sensor modalities like an

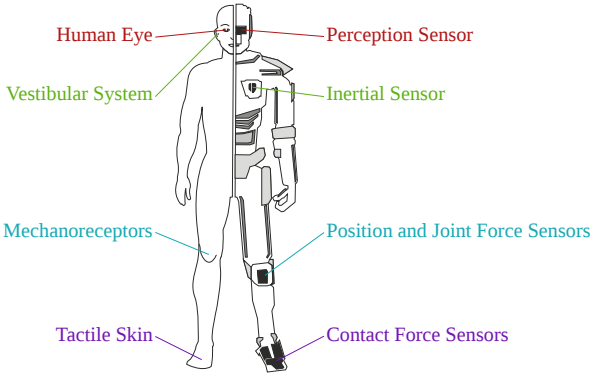

Figure 6.1: Many robotic sensors have a human counterpart.

Inertial Measurements Unit (IMU) or a camera.

2 Modeling

Before discussing the modeling in more detail a brief introduction into the employed notation and convention is provided. Since one asset of legged robots is to deal with more complex non-planar environment it is of high importance to employ 3D models. This requires the proper handling of 3D rotations including an appropriate parametrization and representation of local differences. Based on this, legged robot specific modeling including contact modeling and system dynamics will be discussed in the second part of this section. This will also cover a brief overview of the most commonly employed sensors.

2.1 Employed Notation and Conventions

Throughout this chapter vectors will be denoted by small bold face letters, e.g. $\boldsymbol{v}$, and matrices by capital bold face letters, e.g. $\boldsymbol{M}$. Coordinate frames and physical points in space will both be referred to by regular capital letters, e.g. B. Thus if a coordinate frame is associated with a specific origin the same capital letter can be used for representing both. In order to represent the coordinate vector from point A to point B expressed in the coordinate frame C the term ${}_C\boldsymbol{r}_{AB}$ will be used. The following coordinate frames will be recurring throughout this chapter:

- I: Inertial coordinate frame,
- B: Moving coordinate frame associated with the robot's main body or a mounted sensor,
- F_i: Coordinate frame attached to foot i (the index is sometimes omitted).

Rotations will be parametrized by unit quaternions using the *Hamilton* convention, but in most cases other parametrizations could be employed. The unit quaternion $\boldsymbol{q}_{AB}$ represents the rotation between coordinate frame B and A. It can be mapped to the corresponding rotation matrix $\boldsymbol{C}(\boldsymbol{q}_{AB}) \in \mathbb{R}^{3\times3}$ which transforms the coordinates of a vector expressed in coordinate frame B to the coordinates of the same vector expressed in coordinate frame A, i.e., ${}_A\boldsymbol{v} = \boldsymbol{C}(\boldsymbol{q}_{AB}){}_B\boldsymbol{v}$. Concatenation of quaternions is from right to left and is not commutative, i.e., $\boldsymbol{q}_{AC} = \boldsymbol{q}_{AB} \otimes \boldsymbol{q}_{BC} \neq \boldsymbol{q}_{BC} \otimes \boldsymbol{q}_{AB}$. A concept for addition, subtraction, and differentiation of 3D rotations [16] is summarized in section 6.

The physical vector $\boldsymbol{v}_B$ denotes the absolute velocity of a point B, i.e., the position of B differentiated with respect to some inertial frame. Similarly $\boldsymbol{a}_B$ denotes the absolute acceleration of a point B. Rotational rates will be referred to by $\boldsymbol{\omega}_{AB}$, which is the rotational rate of coordinate system B with respect to coordinate system A. In some cases further denotations like tildes (measurements) or hats (estimates) are employed if a specific aspect of a certain quantity should be highlighted. The superscript $^\times$ is used to denote the skew symmetric matrix ${}_A\boldsymbol{v}^\times \in \mathbb{R}^{3\times3}$ of a coordinate vector ${}_A\boldsymbol{v} \in \mathbb{R}^3$.

Most algorithms in this chapter employ zero-mean Gaussian noise as underlying stochastic models. A multivariate random vector will be denoted by $\boldsymbol{n}_i$ together with the index i for labeling the noise. This will be associated with a covariance matrix $\boldsymbol{R}_i$ and can thus be written as $\boldsymbol{n}_i \sim \mathcal{N}(0, \boldsymbol{R}_i)$.

2.2 General Legged System Modeling

Dynamic System Description

In general, the multi-body equations of motion for a legged robot with n degrees of freedom can be written as follow:

$$\boldsymbol{M}(\boldsymbol{\theta})\dot{\boldsymbol{u}} + \boldsymbol{b}(\boldsymbol{\theta}, \boldsymbol{u}) + \boldsymbol{g}(\boldsymbol{\theta}) + \boldsymbol{J}_c^T(\boldsymbol{\theta})\boldsymbol{F}_c = \boldsymbol{S}^T\boldsymbol{\tau}, \tag{6.1}$$

with $\boldsymbol{\theta}$ denoting the generalized coordinates of the system, $\boldsymbol{F}_c \in \mathbb{R}^m$ the contact forces acting on the system, and $\boldsymbol{\tau} \in \mathbb{R}^l$ the joint forces. While $\boldsymbol{\theta}$ may be over-parametrized (e.g. it may contain quaternions), its derivatives, the generalized velocities $\boldsymbol{u}$ and the generalized accelerations $\dot{\boldsymbol{u}}$, are required to be in the n dimensional vector space $\mathbb{R}^n$. The remaining terms are the inertia matrix $\boldsymbol{M}(\boldsymbol{\theta}) \in \mathbb{R}^{n\times n}$, the combined Coriolis and centrifugal term $\boldsymbol{b}(\boldsymbol{\theta}, \boldsymbol{u}) \in \mathbb{R}^n$, the gravity term $\boldsymbol{g}(\boldsymbol{\theta})$, the contact Jacobian $\boldsymbol{J}_c(\boldsymbol{\theta}) \in \mathbb{R}^{m\times n}$, and the selection matrix $\boldsymbol{S} \in \mathbb{R}^{l\times n}$. Furthermore, assuming no slippage, the contact condition provides the following constraint:

$$\boldsymbol{J}_c(\boldsymbol{\theta})\boldsymbol{u} = \boldsymbol{0}, \tag{6.2}$$

which is basically requiring the velocity of the contact to be zero. In the context of dynamics, the time-differentiated form of this equation is often employed:

$$\dot{\boldsymbol{J}}_c(\boldsymbol{\theta})\boldsymbol{u} + \boldsymbol{J}_c(\boldsymbol{\theta})\dot{\boldsymbol{u}} = \boldsymbol{0}. \tag{6.3}$$

In 3D space the generalized coordinates $\boldsymbol{\theta}$ typically include the pose of the free floating base B with respect to an inertial frame I. The pose is composed of position

and attitude and can be parametrized by $({}_I\boldsymbol{r}_{IB}, \boldsymbol{q}_{BI})$. The rest of the generalized coordinates describe the internal configuration of the robot including quantities like joint angles or link lengths. This last part of the generalized coordinate is referred to as $\boldsymbol{\alpha}$. The generalized coordinates are thus composed of:

$$\boldsymbol{\theta} = ({}_I\boldsymbol{r}_{IB}, \boldsymbol{q}_{BI}, \boldsymbol{\alpha}). \tag{6.4}$$

The above equations of motion often involve analytically complex terms and can be expensive to compute. They provide a relation between the minimal coordinates, the joint forces and the contact forces and can be employed in different ways. For instance, in a simulation scenario all quantities are given except for the generalized accelerations $\dot{\boldsymbol{u}}$ and the contact forces $\boldsymbol{F}_c$. For motion estimation, on the other hand, the set of given quantities strongly depends on the available sensor measurements and on the set of assumptions that can be made. E.g. a specific legged robot might be equipped with joint encoders and load cells for measuring joint positions and forces, but it could lack the ability to measure the contact forces. The contact forces will thus be unknowns in the equations of motion and would have to be either eliminated analytically or solved for. As will be shown in section 4.4, there is a simple way to eliminate the contact forces by employing a null-space projection.

Contact Modeling

Contacts are often modeled as unilateral (no ground penetration) and as non-slipping (infinite friction). Therefore contact forces can only arise along forbidden directions of the contact. These directions coincide with the gradients of the vector of stacked contact constraints $\boldsymbol{c}(\boldsymbol{\theta}) = \boldsymbol{c}_0$:

$$\boldsymbol{J}_c(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} \boldsymbol{c}(\boldsymbol{\theta}). \tag{6.5}$$

Two commonly used contact models are the *point foot* model and the *flat foot* model (see fig. 6.2). A *point foot* is modeled by enforcing a point constraint at the location of contact with the ground. In 3D the dimension of the *point foot* constraint is 3. Mathematically it can be expressed as equality constraint on the location of the contact point F w.r.t. to the inertial coordinate frame I:

$$\boldsymbol{c}(\boldsymbol{\theta}) = {}_I\boldsymbol{r}_{IB} + \boldsymbol{C}(\boldsymbol{q}_{BI})^T {}_B\boldsymbol{r}_{BF}(\boldsymbol{\alpha}) \overset{!}{=} {}_I\boldsymbol{r}^*_{IF}. \tag{6.6}$$

The vector ${}_B\boldsymbol{r}_{BF}(\boldsymbol{\alpha})$ represents the location of the contact point F w.r.t. the body coordinate frame B and is a function of the robot's internal configuration $\boldsymbol{\alpha}$. The coordinate vector ${}_I\boldsymbol{r}^*_{IF}$ represents the stationary location of the foot while in contact with the ground. Using the identities in section 6, the Jacobian of eq. (6.6) can be written as follows:

$$\boldsymbol{J}_c(\boldsymbol{\theta}) = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{C}(\boldsymbol{q}_{BI})^T {}_B\boldsymbol{r}_{BF}(\boldsymbol{\alpha})^\times & \boldsymbol{C}(\boldsymbol{q}_{BI})^T {}_B\boldsymbol{J}_{BF}(\boldsymbol{\alpha}) \end{bmatrix}, \tag{6.7}$$

with the generalized coordinates as in eq. (6.4) and

$${}_B\boldsymbol{J}_{BF}(\boldsymbol{\alpha}) = \frac{\partial}{\partial \boldsymbol{\alpha}} {}_B\boldsymbol{r}_{BF}(\boldsymbol{\alpha}). \tag{6.8}$$

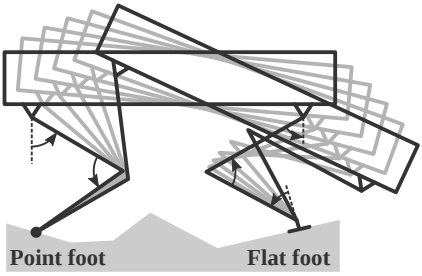

Figure 6.2: Two commonly employed types of foot models. The point foot model requires only a position constraint at the point of contact. The flat foot model also blocks the rotational degree of freedom at the contact point. In the case of fully actuated robots, the flat foot model requires more joints than the point foot model (in the above 2D case 3 instead of 2 actuated joints).

The other type of contact model discussed here is the *flat foot* contact. It can often be found with humanoid robots. In addition to a position constraint it also exhibit a rotational constraint around the foot location as long as proper contact is preserved with the ground. In 3D the total dimension of the *flat foot* constraint is 6. The coordinate system F which is associated with the *flat foot* is assumed to remain stationary while the foot maintains ground contact. This extends the *point foot* model by adding a rotational constraint around the foot. Mathematically this can be expressed as:

$$\boldsymbol{c}(\boldsymbol{\theta}) = \begin{pmatrix} {}_I\boldsymbol{r}_{IB} + \boldsymbol{C}(\boldsymbol{q}_{BI})^T {}_B\boldsymbol{r}_{BF}(\boldsymbol{\alpha}) \\ \boldsymbol{q}_{FB}(\boldsymbol{\alpha}) \otimes \boldsymbol{q}_{BI} \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} {}_I\boldsymbol{r}^*_{IF} \\ \boldsymbol{q}^*_{FI} \end{pmatrix}. \tag{6.9}$$

The quaternion $\boldsymbol{q}_{FB}(\boldsymbol{\alpha})$ parametrizes the rotation between body and foot coordinate frame and can often be computed using a forward kinematics model. The quaternion $\boldsymbol{q}^*_{FI}$ parametrizes the stationary orientation of the foot during contact. Again using the identities in section 6, the Jacobian can be computed and written as:

$$\boldsymbol{J}_c(\boldsymbol{\theta}) = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{C}(\boldsymbol{q}_{BI})^T {}_B\boldsymbol{r}_{BF}(\boldsymbol{\alpha})^\times & \boldsymbol{C}(\boldsymbol{q}_{BI})^T {}_B\boldsymbol{J}_{BF}(\boldsymbol{\alpha}) \\ \boldsymbol{0} & \boldsymbol{C}(\boldsymbol{q}_{FB}(\boldsymbol{\alpha})) & \boldsymbol{J}_{FB}(\boldsymbol{\alpha}) \end{bmatrix}, \tag{6.10}$$

with

$$\boldsymbol{J}_{FB}(\boldsymbol{\alpha}) = \frac{\partial}{\partial \boldsymbol{\alpha}} \boldsymbol{q}_{FB}(\boldsymbol{\alpha}). \tag{6.11}$$

The above contact models assume no slippage at the contact point. While this is often employed as underlying assumption for legged state estimation algorithms, many algorithms are able to intrinsically handle a certain violation of this assumption. Some algorithms implement slippage detection methodologies in order to detect slipping feet and adapt the estimation process accordingly [11, 43, 92, 116].

2.3 Sensor Models

Kinematic Sensors

Kinematic sensors measure the internal configuration of the robot and often directly depend on the corresponding state $\boldsymbol{\alpha}$:

$$\tilde{\boldsymbol{z}}_{kin} = \boldsymbol{f}_{kin}(\boldsymbol{\alpha}, \boldsymbol{n}_\alpha). \tag{6.12}$$

The noise term $\boldsymbol{n}_\alpha$ is often modeled as zero-mean discrete Gaussian noise with covariance $\boldsymbol{R}_\alpha$. In the case the measured quantities are equivalent to $\boldsymbol{\alpha}$, e.g., a robot where every joint is equipped with an encoder, this can be simplified to:

$$\tilde{\boldsymbol{z}}_{kin} = \boldsymbol{\alpha} + \boldsymbol{n}_\alpha. \tag{6.13}$$

In the context of state estimation, the additive Gaussian noise model is often sufficient and more complex noise models are only rarely used.

The typical kinematic sensor is the encoder. Different type of physical principles are employed, e.g. mechanical, optical or magnetic principles [121]. Furthermore they can be manufactured as linear or rotational encoders. The linear encoder measures the length or extension of a linear joint and the rotational encoder measures the angle of a rotation joint. An important differentiation must be made between *relative* and *absolute* encoders. *Relative* encoders count so-called "ticks" and thereby compute the difference between the actual encoder position and some reference position (e.g. the position the sensor was initialized at). In contrast to this, absolute encoders measure the absolute encoder deflection which is encoded at the current position. They have the advantage that after each re-start they yield the same encoder output if measuring the same deflection. However, absolute encoder typically tend to be a bit more expensive when compared to relative encoders with the same resolution.

Force Sensors and Contact Detection

Force sensors can be employed to measure internal forces such as joint forces or to measure external forces such as contact forces. Often, they can be modeled similarly to kinematics sensors:

$$\tilde{\boldsymbol{z}}_{dyn} = \boldsymbol{f}_{dyn}(\boldsymbol{\tau}, \boldsymbol{F}_c, \boldsymbol{n}_{dyn}). \tag{6.14}$$

where $\boldsymbol{\tau}$ and $\boldsymbol{F}_c$ represents the internal forces and contact forces (see section 2.2) and $\boldsymbol{n}_{dyn}$ is zero-mean discrete Gaussian noise with covariance $\boldsymbol{R}_{dyn}$.

Again, there are many different types of sensors including optical, resistive, capacitive, or piezoelectric [121]. One difficulty involves the calibration of the force sensor, which due to the lack of precise reference has often to be carried out by the manufacturer and cannot be corrected afterwards. This is one reason why force sensors represent a less reliable source of information when it comes to state estimation. Consequently force sensors are often only used for estimating the contact state, i.e., whether a foot is in contact with the ground or not. If a contact force is not directly measured by a force sensor, it can also be indirectly estimated by using the equation of motion (eq. (6.1)).

Inertial Measurement Unit

Inertial measurements units (IMUs) are sensor devices which contain at least a gyroscope and an accelerometer (in general both 3-axis). For the sake of simplicity the coordinate frame that is attached to the IMU and in which the IMU measurements are expressed is also denoted by B. The gyroscope measures the rotational rate of the device ${}_B\tilde{\boldsymbol{\omega}}_{IB}$. The accelerometer measures its proper acceleration ${}_B\tilde{\boldsymbol{f}}_{IB}$ which is the superposition of the actual acceleration of the device and the gravitational acceleration. Other sensor modalities which are sometimes included in IMUs are magnetometers, barometers, or GPS.

Most reasonable IMUs have an internal calibration routine which is responsible for reducing nonlinear effects and performing temperature compensation of the measured rotational rate and proper acceleration. But even with these measurement corrections, remaining measurement noise and bias are never entirely avoidable. This can be modeled in different ways, but the most common approach is to model it as additive noise and bias on the IMU outputs (an in-depth discussion can be found in [105]). The resulting stochastic model can be written as:

$$ {}_B\tilde{\boldsymbol{f}}_{IB} = \boldsymbol{C}(\boldsymbol{q}_{BI})({}_I\boldsymbol{a}_B - {}_I\boldsymbol{g}) + \boldsymbol{b}_f + \boldsymbol{n}_f, \tag{6.15} $$

$$ \dot{\boldsymbol{b}}_f = \boldsymbol{n}_{bf}, \tag{6.16} $$

$$ {}_B\tilde{\boldsymbol{\omega}}_{IB} = {}_B\boldsymbol{\omega}_{IB} + \boldsymbol{b}_\omega + \boldsymbol{n}_\omega, \tag{6.17} $$

$$ \dot{\boldsymbol{b}}_\omega = \boldsymbol{n}_{b\omega}, \tag{6.18} $$

where ${}_I\boldsymbol{g}$ is the gravitational acceleration expressed in the inertial frame I. The additive white Gaussian noise processes $\boldsymbol{n}_f, \boldsymbol{n}_{bf}, \boldsymbol{n}_\omega, \boldsymbol{n}_{b\omega}$ are added to the above terms in order to model the continuous time noise that affects the IMU outputs. This is a common modeling approach for IMU measurements, where the corresponding covariance parameters $\boldsymbol{R}_f, \boldsymbol{R}_{bf}, \boldsymbol{R}_\omega, \boldsymbol{R}_{b\omega}$ can be derived by looking at the corresponding Allan variance plots [32].

Especially in the consumer market IMUs have seen a huge upswing in the past few years, providing increasingly performing devices at low prices. IMUs can be assigned to different grading categories depending on their gyroscope bias stability parameter bs, which specifies how well the gyroscope bias can possibly be estimated. A lower bound for the bias stability parameter is given by $bs > 0.5(\sqrt{\boldsymbol{R}_\omega \cdot Hz} + \sqrt{\boldsymbol{R}_{b\omega}/Hz})$ and can be used as alternative grading criteria. A possible choice of grading categories is given by: tactical grade ($bs > 0.1°/hr$), navigation grade ($0.1°/hr > bs > 0.0001°/hr$), and strategical grade ($0.0001°/hr > bs$).

There are algorithms which directly estimate the attitude of the IMU based on the accelerometer and gyroscope measurements only. To this end, an often employed assumption is that the mean acceleration over time is zero. This allows to correct the integrated gyroscope values and to obtain a relatively stable longterm attitude estimation. One critical part of this approach is given by the gyroscope bias which has to be estimated online. Since an erroneous bias estimate gets directly integrated into the attitude estimation it has a strong influence on its accuracy. The quality of the bias estimate itself strongly depends on the bias stability parameter bs, which explain why this parameter is one of the main grading criteria for IMUs.

Environment Perception Sensors

To enhance state estimation and in particular to provide means of determining absolute positions and yaw orientations, legged robots are sometimes equipped with cameras or laser range finders (Lidar). Both sensors require relatively complex algorithms to calculate the robot state from image data or laser scans. Cameras are lightweight, cheap and energy efficient sensors that provide high density information on the surrounding. Their main draw-back is that cameras tend to be sensitive to bad lighting and fast motions which can quickly deteriorate the quality of the output. There is a vast amount of state-of-the art visual odometry approaches which show very impressive results in term of localization accuracy [33, 40, 131]. Additional robustness can be gained by employing a visual-inertial approach which integrates inertial measurements into the visual ego-motion estimation algorithm [14, 84].

Range sensors, on the other hand, are relatively heavy, power hungry, and expensive. They also require additional processing if the robot is in motion during the scanning. In comparison to camera image data, they provide depth measurements which are less depending on illumination and motion of the system. Localization is typically retrieved by point cloud matching, whereby iterative closest point (ICP) algorithms are very popular [110].

3 Sensor Fusion

Key element for robust and reliable state estimation is proper fusion of the different sensor modalities. While there exists a vast amount of possible optimization and filtering techniques, Kalman filter based algorithms remain the most employed. In the context of this section, the basic concepts of general Kalman filtering are discussed using the example of the Extended Kalman Filter (EKF). A comprehensive introduction and overview on Kalman filters, including information on computational costs, can be found in [52].

A Kalman filter is a stochastic filter that fuses the information from different sensor modalities [77]. In the case of a linear system, a Kalman filter provides the optimal estimate of an information fusion problem in terms of minimum mean square error of the estimate. For general nonlinear systems, no guarantee for optimality or convergence can be given.

One of the key aspect of a Kalman filter is the choice and parametrization of its state $\boldsymbol{x}$, with which a proper system model is formulated. The discrete-time model is of the following form:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \tilde{\boldsymbol{z}}_{f,k}, \boldsymbol{n}_{f,k}), \tag{6.19}$$

$$\boldsymbol{y}_{k+1} = \boldsymbol{h}(\boldsymbol{x}_{k+1}, \tilde{\boldsymbol{z}}_{h,k+1}, \boldsymbol{n}_{h,k+1}), \tag{6.20}$$

where $\boldsymbol{n}_{f,k} \sim \mathcal{N}(0, \boldsymbol{R}_{f,k})$ and $\boldsymbol{n}_{h,k+1} \sim \mathcal{N}(0, \boldsymbol{R}_{h,k+1})$ are discrete Gaussian noise. Only with an appropriate choice of the state $\boldsymbol{x}$ it is possible to formulate a process model $\boldsymbol{f}$ and an update model $\boldsymbol{h}$ such that the information contained in the measurements $\tilde{\boldsymbol{z}}_{f,k}$ and $\tilde{\boldsymbol{z}}_{h,k}$ can be properly leveraged into the state estimation. In the selected formulation, the update model directly outputs the Kalman innovation term $\boldsymbol{y}_{k+1}$, which is a slightly more generalized form of the regular measurement model

$\tilde{\boldsymbol{z}}_{h,k+1} = \boldsymbol{h}(\boldsymbol{x}_{k+1}, \boldsymbol{n}_{h,k+1})$. As a general design rule, every sensor measurement that is used during the update step must yield an innovation term that depends on the measurement itself and the filter state. Kalman filters always keep track of the covariance of the filter state (or some related quantity):

$$\boldsymbol{P}_k = \mathrm{cov}(\boldsymbol{x}_k). \tag{6.21}$$

The EKF makes use of the same equations as the regular linear Kalman filter except for replacing the linear system matrices with the Jacobians of the above nonlinear system. In the following the a-priori estimated filter state (i.e. before the inclusion of update measurements) will be decorated with a minus superscript $^-$ and the a-posteriori (i.e. after the inclusion of update measurements) will be decorated with a plus superscript $^+$. The current best estimate of the state is used when evaluating the Jacobians:

$$\boldsymbol{F}_k := \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}_k, \tilde{\boldsymbol{z}}_{f,k}, \boldsymbol{n}_{f,k})}{\partial \boldsymbol{x}_k}\right|_{(\hat{\boldsymbol{x}}_k^+, \tilde{\boldsymbol{z}}_{f,k}, \boldsymbol{0})}, \tag{6.22}$$

$$\boldsymbol{G}_k := \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}_k, \tilde{\boldsymbol{z}}_{f,k}, \boldsymbol{n}_{f,k})}{\partial \boldsymbol{n}_{f,k}}\right|_{(\hat{\boldsymbol{x}}_k^+, \tilde{\boldsymbol{z}}_{f,k}, \boldsymbol{0})}, \tag{6.23}$$

$$\boldsymbol{H}_{k+1} := \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x}_{k+1}, \tilde{\boldsymbol{z}}_{h,k+1}, \boldsymbol{n}_{h,k+1})}{\partial \boldsymbol{x}_{k+1}}\right|_{(\hat{\boldsymbol{x}}_{k+1}^-, \tilde{\boldsymbol{z}}_{h,k+1}, \boldsymbol{0})}, \tag{6.24}$$

$$\boldsymbol{J}_{k+1} := \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x}_{k+1}, \tilde{\boldsymbol{z}}_{h,k+1}, \boldsymbol{n}_{h,k+1})}{\partial \boldsymbol{n}_{h,k+1}}\right|_{(\hat{\boldsymbol{x}}_{k+1}^-, \tilde{\boldsymbol{z}}_{h,k+1}, \boldsymbol{0})}. \tag{6.25}$$

The recursive filter equations for predicting the state and covariance matrix are:

$$\hat{\boldsymbol{x}}_{k+1}^- = \boldsymbol{f}(\hat{\boldsymbol{x}}_k^+, \tilde{\boldsymbol{z}}_{f,k}, \boldsymbol{0}), \tag{6.26}$$

$$\boldsymbol{P}_{k+1}^- = \boldsymbol{F}_k \boldsymbol{P}_k^+ \boldsymbol{F}_k^T + \boldsymbol{G}_k \boldsymbol{R}_{f,k} \boldsymbol{G}_k^T. \tag{6.27}$$

For the update step, the recursive filter equations have the form:

$$\boldsymbol{S}_{k+1} = \boldsymbol{H}_{k+1} \boldsymbol{P}_{k+1}^- \boldsymbol{H}_{k+1}^T + \boldsymbol{J}_{k+1} \boldsymbol{R}_{h,k+1} \boldsymbol{J}_{k+1}^T, \tag{6.28}$$

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1}^- \boldsymbol{H}_{k+1}^T \boldsymbol{S}_{k+1}^{-1}, \tag{6.29}$$

$$\boldsymbol{P}_{k+1}^+ = (\boldsymbol{I} - \boldsymbol{K}_{k+1} \boldsymbol{H}_{k+1}) \boldsymbol{P}_{k+1}^-, \tag{6.30}$$

$$\Delta \boldsymbol{x}_{k+1} = \boldsymbol{K}_{k+1} \boldsymbol{h}(\hat{\boldsymbol{x}}_{k+1}^-, \tilde{\boldsymbol{z}}_{h,k+1}, \boldsymbol{0}), \tag{6.31}$$

$$\hat{\boldsymbol{x}}_{k+1}^+ = \hat{\boldsymbol{x}}_{k+1}^- \boxplus \Delta \boldsymbol{x}_{k+1}. \tag{6.32}$$

In the last equation the boxplus operator $\boxplus$ is used in order to highlight that if the state $\boldsymbol{x}$ contains some special over-parametrized quantities, such as unit quaternions, the correction $\Delta \boldsymbol{x}_{k+1}$ must be applied using a proper operation (please see section 6 or refer to [16, 59] for more details).

The main reason why the legged state estimation community is mostly using Kalman filters is its simple implementation, its low memory usage due to its recursive formulation, as well as the undelayed estimation (i.e. a full update is generate after every step). Another possibility of information fusion is based on maximum likelihood optimization over the full data (including past data). This often provides more accurate estimates but can be computationally more expensive and can come with an increased time delay until the estimates are available. There are also many intermediate or combined approaches, such as sliding window estimation approaches [84] or other forms of delayed filtering, which are becoming increasingly popular (especially in the computer vision community).

4 Approaches

4.1 Overview

In contrast to wheeled, flying, or swimming machines, legged robots uniquely feature intermittent ground contacts as additional source of information. Almost all approaches employ an underlying no-slippage assumption for the feet in contact with the ground, whereby some can intrinsically account for a certain amount of slippage. While originally designed for either *point feet* or *flat feet* contacts, most methods can be adapted to other types of feet. The different approaches can be characterized based on the employed *technologies*:

- *Matching technologies* compare two subsequent feet configuration and thereby compute the incremental motion. These legged odometry methods are relatively easy to implement. Additional information like e.g. knowledge about the ground plane can be leveraged by direct integration into the odometry.

- *Fusion technologies* combine the information from the intermittent contact with further sensor modalities (often an IMU). They mostly avoid the use of further assumption like even floors. Due to information fusion, these methods are known to provide robust and accurate state estimation. The underlying fusion algorithms often increase complexity and the computational costs.

- *Technologies for leveraging dynamics* exploit available force measurements together with dynamic models in order to improve ego-motion estimation. Due to their complexity and the rather high noise amplitudes of the force measurements they have not been used widely. Still, force measurements can provide an alternate method for estimating the gravity direction and thereby replace or improve the attitude estimation which usually strongly relies on the IMU measurements.

4.2 Matching Technology

Single Foot Matching

Single foot matching approaches make use of the kinematic measurements corresponding to a single foot which remains in contact with the ground between two successive timesteps and directly estimate the incremental motion of the robot's main body. The

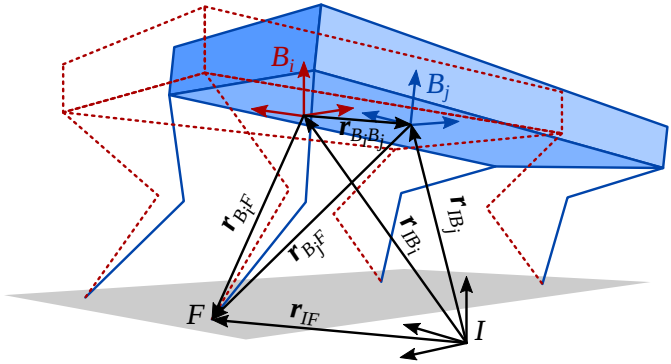

Figure 6.3: Contact point matching for two stances that share the same foothold F. The unknown incremental translation can be estimated by using the triangle $\boldsymbol{r}_{B_iF}$, $\boldsymbol{r}_{B_jF}$, $\boldsymbol{r}_{B_iB_j}$ together with the forward kinematics model of the robot. All terms need to be represented in the same coordinate frame which requires the relative rotation $\boldsymbol{q}_{B_iB_j}$ between the coordinate frame B_i and B_j. This could be obtained from on-board rotational rate measurements.

approach is not really limited to successive timesteps and can be applied between arbitrary robot stances sharing the same ground contact. For a *point foot* model eq. (6.6) must be fulfilled for every timestep sharing the same point contact F. For two such timesteps i and j, this means that the incremental translation between both stances can be written as

$$ {}_{B_i}\boldsymbol{r}_{B_iB_j} = {}_{B_i}\boldsymbol{r}_{B_iF} - \boldsymbol{C}(\boldsymbol{q}_{B_iB_j})_{B_j}\boldsymbol{r}_{B_jF}, \tag{6.33} $$

where ${}_{B_i}\boldsymbol{r}_{B_iF} := {}_{B}\boldsymbol{r}_{BF}(\boldsymbol{\alpha}_i)$ and ${}_{B_j}\boldsymbol{r}_{B_jF} = {}_{B}\boldsymbol{r}_{BF}(\boldsymbol{\alpha}_j)$ are given by the forward kinematics of the corresponding leg at timestep i and j, and where $\boldsymbol{q}_{B_iB_j}$ is required in order to account for the rotational motion of the main body. The quantities are illustrated in fig. 6.3.

The rotational term $\boldsymbol{q}_{B_iB_j}$ can be neglected if the robot has only little rotational motion between both timesteps, i.e., if the time difference is very small or if the motion of the robot does not exhibit quick rotations. If this is not the case the rotational term has to be estimated by some other means. Often rotational rates measurements from an on-board gyroscope represent the simplest and most accurate way of getting such an incremental rotational motion estimate. Furthermore, since the incremental translation ${}_{B_i}\boldsymbol{r}_{B_iB_j}$ is given in body coordinates, an external attitude estimation is indispensable if the robot should estimate its global trajectory. In some cases the incremental translation is also employed in its differentiated form as velocity measurement [35, 92].

An alternative is present if a *flat foot* contact model can be applied. In this case eq. (6.9) must be fulfilled for every timestep sharing the same *flat foot* contact with

associated coordinate frame F. The second part of the constraint can directly be used for estimating the incremental rotation:

$$\boldsymbol{q}_{B_iB_j} = \boldsymbol{q}_{FB_i}^{-1} \otimes \boldsymbol{q}_{FB_j}, \tag{6.34}$$

with $\boldsymbol{q}_{FB_i} := \boldsymbol{q}_{FB}(\boldsymbol{\alpha}_i)$ and $\boldsymbol{q}_{FB_j} := \boldsymbol{q}_{FB}(\boldsymbol{\alpha}_j)$. This means that for each *flat foot* that is in contact with the ground the full incremental motion of the main body can be estimate by:

$${}_{B_i}\boldsymbol{r}_{B_iB_j} = {}_{B_i}\boldsymbol{r}_{B_iF} - \boldsymbol{C}(\boldsymbol{q}_{FB_i}^{-1} \otimes \boldsymbol{q}_{FB_j}){}_{B_j}\boldsymbol{r}_{B_jF}, \tag{6.35}$$

$$\boldsymbol{q}_{B_iB_j} = \boldsymbol{q}_{FB_i}^{-1} \otimes \boldsymbol{q}_{FB_j}. \tag{6.36}$$

This is a widely used odometry concept for humanoid robots since as long as a contact with the ground is available an incremental motion can be estimated. However, due to slippage or model inaccuracies this approach is not very accurate and prone to drift. Also, if no accelerometer or force data is used there is no way how the gravity direction can be estimated reliably.

Multiple Foot Matching

Taking more than one ground contact into account enables the design of potentially more accurate odometry approaches. Furthermore, if enough (at least three non-colinear) contacts are available, the incremental rotation can also be estimated with point contacts only. One of the first such approach was demonstrated by Roston and Krotkov [116] on their Ambler hexapod. They formulated an optimization which minimizes the error between matching ground contact in order to estimate the incremental motion (translation and rotation).

The basic idea is to form an error term out of the identity in eq. (6.33) for every of the N foot point F_k in contact with the ground:

$$\boldsymbol{e}_k = {}_{B_i}\boldsymbol{r}_{B_iB_j} - {}_{B_i}\boldsymbol{r}_{B_iF_k} + \boldsymbol{C}(\boldsymbol{q}_{B_iB_j}){}_{B_j}\boldsymbol{r}_{B_jF_k}. \tag{6.37}$$

Using the abbreviations $\boldsymbol{t} = {}_{B_i}\boldsymbol{r}_{B_iB_j}$, $\boldsymbol{q} = \boldsymbol{q}_{B_iB_j}$, $\boldsymbol{a}_k = {}_{B_i}\boldsymbol{r}_{B_iF_k}$, and $\boldsymbol{b}_k = {}_{B_j}\boldsymbol{r}_{B_jF_k}$ the term can be rewritten as

$$\boldsymbol{e}_k(\boldsymbol{t}, \boldsymbol{q}) = \boldsymbol{t} - \boldsymbol{a}_k + \boldsymbol{C}(\boldsymbol{q})\boldsymbol{b}_k. \tag{6.38}$$

This can now be transformed to a quaternion form by mapping a 3D vector $\boldsymbol{t}$ to a purely virtual quaternion $\bar{\boldsymbol{t}} = \boldsymbol{S}^T\boldsymbol{t}$ with the selection matrix

$$\boldsymbol{S} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{6.39}$$

The selection matrix can also be used to compute a rotated coordinate vector as $\boldsymbol{C}(\boldsymbol{q})\boldsymbol{b} = \boldsymbol{S}\left(\boldsymbol{q} \otimes \boldsymbol{S}^T\boldsymbol{b} \otimes \boldsymbol{q}^{-1}\right)$. Consequently, the quaternion form of the error term in eq. (6.38) can be expressed as:

$$\bar{\boldsymbol{e}}_k(\boldsymbol{t}, \boldsymbol{q}) = \left(\bar{\boldsymbol{t}} - \bar{\boldsymbol{a}}_k + \boldsymbol{q} \otimes \bar{\boldsymbol{b}}_k \otimes \boldsymbol{q}^{-1}\right) \otimes \boldsymbol{q}, \tag{6.40}$$

$$= \bar{\boldsymbol{t}} \otimes \boldsymbol{q} - \bar{\boldsymbol{a}}_k \otimes \boldsymbol{q} + \boldsymbol{q} \otimes \bar{\boldsymbol{b}}_k, \tag{6.41}$$

where $\bar{\boldsymbol{t}} = \boldsymbol{S}^T\boldsymbol{t}$, $\bar{\boldsymbol{a}}_k = \boldsymbol{S}^T\boldsymbol{a}_k$, and $\bar{\boldsymbol{b}}_k = \boldsymbol{S}^T\boldsymbol{b}_k$ are pure imaginary quaternions and where a right multiplication by $\boldsymbol{q}$ was applied (this does not influence the norm of the error term because $\boldsymbol{q}$ is a unit quaternion).

One advantage of the quaternion parametrization is that the obtained error term $\bar{\boldsymbol{e}}_k(\boldsymbol{t}, \boldsymbol{q})$ is linear in $\boldsymbol{q}$ and equals zero if the contact constraint is fulfilled. This is enabled by the linear nature of the quaternion multiplication which can be represented as

$$\boldsymbol{q} \otimes \boldsymbol{p} = \boldsymbol{L}(\boldsymbol{q})\boldsymbol{p} = \boldsymbol{R}(\boldsymbol{p})\boldsymbol{q}, \tag{6.42}$$

where the matrices $\boldsymbol{L}(\boldsymbol{q}) \in \mathbb{R}^{4\times4}$ and $\boldsymbol{R}(\boldsymbol{p}) \in \mathbb{R}^{4\times4}$ can be expressed as linear maps of the corresponding unit quaternion.

Constructing a least square optimization based on multiple such error terms and including the Lagrangian term (with Lagrangian multiplier λ) for taking into account the unit norm constraint of the quaternion $\boldsymbol{q}$ yields

$$\min_{\boldsymbol{t},\boldsymbol{q},\lambda} \sum_k \bar{\boldsymbol{e}}_k^T(\boldsymbol{t}, \boldsymbol{q})\bar{\boldsymbol{e}}_k(\boldsymbol{t}, \boldsymbol{q}) + \lambda(\boldsymbol{q}^T\boldsymbol{q} - 1). \tag{6.43}$$

It can be shown (see section 7) that the solution to this problem is given by the following Eigenvalue problem for $\boldsymbol{q}$:

$$\boldsymbol{B}\boldsymbol{q} - \lambda\boldsymbol{q} = \boldsymbol{0}, \tag{6.44}$$

with

$$\boldsymbol{B} = \sum_k (\boldsymbol{A} - \boldsymbol{A}_k)(\boldsymbol{A} - \boldsymbol{A}_k) = \sum_k \boldsymbol{A}_k\boldsymbol{A}_k - \boldsymbol{A}\boldsymbol{A}, \tag{6.45}$$

$$\boldsymbol{A} = \sum_k \boldsymbol{A}_k, \quad \boldsymbol{A}_k = \boldsymbol{L}(\bar{\boldsymbol{a}}_k) - \boldsymbol{R}(\bar{\boldsymbol{b}}_k). \tag{6.46}$$

The corresponding solution for the incremental translation $\boldsymbol{t}$ is given by the vector that maps the mean of the rotated $\boldsymbol{b}_k$ to the mean of $\boldsymbol{a}_k$:

$$\boldsymbol{t} = \boldsymbol{a} - \boldsymbol{C}(\boldsymbol{q})\boldsymbol{b}, \quad \boldsymbol{a} = \frac{\sum_k \boldsymbol{a}_k}{N}, \quad \boldsymbol{b} = \frac{\sum_k \boldsymbol{b}_k}{N}. \tag{6.47}$$

Solving this Eigenvalue problem thus provides a solution for the incremental motion $\boldsymbol{t}$ and $\boldsymbol{q}$ without the requirement for additional sensors as long as enough ground contacts are available. This approach has often been applied on hexapods using a tripod gait [43, 51, 87, 116].

The presented result for point feet can be adapted to flat feet and hence to account for rotational constraints. The additional rotational error terms can directly be derived from eq. (6.34):

$$\boldsymbol{e}_{r,k} = \boldsymbol{S}\left(\boldsymbol{q}_{F_kB_j} \otimes \boldsymbol{q}_{B_iB_j} \otimes \boldsymbol{q}_{F_kB_i}^{-1}\right) = \boldsymbol{S}\boldsymbol{L}(\boldsymbol{q}_{F_kB_j})\boldsymbol{R}(\boldsymbol{q}_{F_kB_i}^{-1})\boldsymbol{q}_{B_iB_j}. \tag{6.48}$$

Merging this into the previous optimization yields

$$\min_{\bar{\boldsymbol{t}},\boldsymbol{q},\lambda} \sum_k \boldsymbol{e}_k^T(\bar{\boldsymbol{t}},\boldsymbol{q})\boldsymbol{e}_k(\bar{\boldsymbol{t}},\boldsymbol{q}) + \sum_k \boldsymbol{e}_{r,k}^T(\boldsymbol{q})\boldsymbol{e}_{r,k}(\boldsymbol{q}) + \lambda(\boldsymbol{q}^T\boldsymbol{q}-1), \tag{6.49}$$

which can be shown to have the same solution as the *point foot* problem except for an additive term in the Eigenproblem matrix:

$$\boldsymbol{B} = \sum_k \boldsymbol{A}_k\boldsymbol{A}_k - \boldsymbol{A}\boldsymbol{A} + \sum_k \boldsymbol{R}(\boldsymbol{q}_{F_kB_i})\boldsymbol{q}_{F_kB_j}^{-1}\boldsymbol{q}_{F_kB_j}^{-T}\boldsymbol{R}(\boldsymbol{q}_{F_kB_i}^{-1}). \tag{6.50}$$

The interesting aspect of this combination is that the estimation of the incremental rotation $\boldsymbol{q}$ draws from the position constraint (eq. (6.37)) and the rotation constraint (eq. (6.48)) simultaneously. Especially for humanoids this can be of interest since typically the rotational constraint of the *flat foot* tends to be only weakly satisfied and that further information from the location of the feet can be beneficial.

There are many adaptation to the above approaches. Often they can be derived by directly adapting the formulation of the optimization problem. One important extension is to integrate different weighting into the error terms. For instance, a foot which is suspected to slip can be down-weighted in such a manner that the estimation process relies more on the other feet in contact [43]. Also, when combining position and rotational constraints in a joint optimization (as in eq. (6.49)) the use of weighting is often very important due to the different units of the involved error terms.

4.3 Sensor Fusion for Legged Robots

Sensor fusion approaches *tightly* or *loosely* fuse the kinematic constraints that are used in matching approaches with additional sensor modalities. The term "*tightly*" as opposed to "*loosely*" means that the different sensor measurements are combined in a non-processed form (see fig. 6.4). For instance, for IMU and kinematic measurements this means that the proper acceleration, rotational rate, and joint encoder measurements are directly fused together in order to obtain a single ego-motion estimation. In contrast to this, a loosely coupled approach would first produce ego-motion estimates from the single sensor modalities and then combine them to one. The *tightly* coupled approaches typically have the advantage that better measurement models are applicable and that superfluous assumptions can be avoided.

The following section outlines the *tight* combination of kinematics measurements with IMU readings, the most widely used sensor in legged robots. Inclusion of further sensor modalities is illustrated in the section afterwards. Measurements fusion from different sensor will be discussed using the example of the EKF. Most approaches however could easily be transfered to other types of Kalman filters or to batch optimization approaches. While every Kalman filter setup can be formulated as a batch optimization, this is not necessarily true for the other way around.

Fusion of IMU and Kinematic Measurements

The basic concept is to tightly combine the IMU readings with the kinematic sensing respectively the contact constraints. To this end, IMU measurements are very often

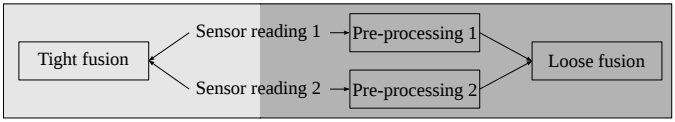

Figure 6.4: A thigh sensor fusion directly employs the sensor measurements in order to estimate a unified output. This is in contract to loose sensor fusion which pre-processes the sensor measurements into intermediate quantities before merging them together. While loose fusion might provide a better overview of the process and can be faster, the tight fusion approach is stochastically more accurate and yields better results.

integrated in the process model of the Kalman filter. This has the advantage that a reduced filter state can be employed where acceleration and rotational rates do not necessarily have to be included.

One technique to integrate kinematic measurements, is to formulate an update model that assumes stationary contact points and penalizes every deviation from it. This can be achieved by extending the filter state to contain the 3D location of the N contact points [10]:

$$\begin{aligned}\boldsymbol{x} &:= \left(\boldsymbol{r}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{b}_f, \boldsymbol{b}_\omega, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_N\right), \\ &= \left({}_I\boldsymbol{r}_{IB}, {}_I\boldsymbol{v}_B, \boldsymbol{q}_{BI}, {}_B\boldsymbol{b}_f, {}_B\boldsymbol{b}_\omega, {}_I\boldsymbol{r}_{IF_1}, \ldots, {}_I\boldsymbol{r}_{IF_N}\right),\end{aligned} \tag{6.51}$$

with:

- ${}_I\boldsymbol{r}_{IB}$: position of IMU w.r.t. the inertial frame I,
- ${}_I\boldsymbol{v}_B$: velocity of IMU w.r.t. the inertial frame I,
- $\boldsymbol{q}_{BI}$: attitude of IMU (map from I to B),
- ${}_B\boldsymbol{b}_f$: additive bias on accelerometer (expressed in B),
- ${}_B\boldsymbol{b}_\omega$: additive bias on gyroscope (expressed in B),
- ${}_I\boldsymbol{r}_{IF_i}$: the location of the i^{th} contact point w.r.t. the inertial frame I.

The process model can be formulated based on eqs. (6.15) to (6.18). It is composed of IMU-based kinematics as well as of random walk models for the IMU biases and the foot contact locations:

$$\dot{\boldsymbol{r}} = \boldsymbol{v} + \boldsymbol{n}_v, \tag{6.52}$$

$$\dot{\boldsymbol{v}} = \boldsymbol{C}(\boldsymbol{q})^T({}_B\tilde{\boldsymbol{f}}_{IB} - \boldsymbol{b}_f - \boldsymbol{n}_f) + \boldsymbol{g}, \tag{6.53}$$

$$\dot{\boldsymbol{q}} = {}_B\tilde{\boldsymbol{\omega}}_{IB} - \boldsymbol{b}_\omega - \boldsymbol{n}_\omega, \tag{6.54}$$

$$\dot{\boldsymbol{b}}_f = \boldsymbol{n}_{bf}, \tag{6.55}$$

$$\dot{\boldsymbol{b}}_\omega = \boldsymbol{n}_{b\omega}, \tag{6.56}$$

$$\dot{\boldsymbol{p}}_i = \boldsymbol{n}_{p,i}. \tag{6.57}$$

The terms of the form $\boldsymbol{n}_*$ represent white Gaussian noise with covariance parameter $\boldsymbol{R}_*$. Most noise parameters can be chosen based on the specifications of the IMU. The covariances $\boldsymbol{R}_{p,i}$ represent how much a *point foot* can move while in contact and is a tuning factor which has to be adapted to the platform and its environment. The set of differential equation can be discretized by using a Euler-forward scheme yielding a process model of the form $\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, (\tilde{\boldsymbol{\omega}}_k, \tilde{\boldsymbol{f}}_k), \boldsymbol{n}_*)$. This model is directly integrated in the Kalman filter in order to obtain a prediction of the filter state and the covariance after processing an IMU measurement.

The innovation term for a leg i, which is in contact with the ground, is derived from the kinematic constraint (eq. (6.6)) expressed in the inertial frame I. This depends on the measured joint positions $\tilde{\boldsymbol{\alpha}}$:

$$\begin{aligned} \boldsymbol{y}_{p,i}(\boldsymbol{x}, \tilde{\boldsymbol{\alpha}}) &= {}_I\boldsymbol{r}_{IB} + \boldsymbol{C}(\boldsymbol{q}_{BI})^T {}_B\boldsymbol{r}_{BF_i}(\tilde{\boldsymbol{\alpha}}) - {}_I\boldsymbol{r}_{IF_i} + \boldsymbol{n}_{cp,i}, && (6.58) \\ &= \boldsymbol{r} + \boldsymbol{C}(\boldsymbol{q})^T {}_B\boldsymbol{r}_{BF_i}(\tilde{\boldsymbol{\alpha}}) - \boldsymbol{p}_i + \boldsymbol{n}_{cp,i}. && (6.59) \end{aligned}$$

Additive noise $\boldsymbol{n}_{cp,i} \sim \mathcal{N}(0, \boldsymbol{R}_{cp,i})$ is included in order to account for a certain amount of modeling errors (e.g. ball-shaped feet with rolling motion) and other disturbances. Similar to the prediction noise $\boldsymbol{R}_{p,i}$, the covariance $\boldsymbol{R}_{cp,i}$ is a tuning factor and has to be adapted to the hardware specification and the application scenario.

It is important to understand that the contact points are truly co-estimated in this approach. When a new contact is made with the ground, a new estimated contact location can be initialized using the current estimated location of the main body and the forward kinematics. The corresponding covariance matrix can be initialized by considering the innovation Jacobian and the system noise. After that, every successive kinematic measurement belonging to this contact point can be processed using eq. (6.58). The Kalman filter equation will take care of propagating the information throughout the filter state while properly considering the correlation between the different states. This will induce corrections on the estimated main body motion as well as on the estimated foot points. As soon as a ground contact gets lost, the corresponding contact point is removed from the filter state. This approach does not use any additional assumption on the shape of the floor or the location of the foot points.

An alternative filter design for *point feet* can be derived when differentiating eq. (6.33) and thus integrating the kinematic information on a velocity level [11]. This yields the following innovation term:

$$\begin{aligned} \boldsymbol{y}_{v,i}(\boldsymbol{x}, \tilde{\boldsymbol{\alpha}}, \dot{\tilde{\boldsymbol{\alpha}}}) = \; & \boldsymbol{v} + \boldsymbol{C}(\boldsymbol{q})^T ({}_B\tilde{\boldsymbol{\omega}}_{IB} \times {}_B\boldsymbol{r}_{BF_i}(\tilde{\boldsymbol{\alpha}})) \\ & + \; \boldsymbol{C}^T(\boldsymbol{q}) \frac{\partial}{\partial \boldsymbol{\alpha}} {}_B\boldsymbol{r}_{BF_i}(\tilde{\boldsymbol{\alpha}}) \dot{\tilde{\boldsymbol{\alpha}}} + \boldsymbol{n}_{cv,i}, \quad (6.60) \end{aligned}$$

where $\dot{\tilde{\boldsymbol{\alpha}}}$ is the measured joint velocity and $\boldsymbol{n}_{cv,i}$ discrete Gaussian noise. This basically exploits the fact that the velocity of a foot which is in contact with the ground is zero. While exhibiting slightly higher drift on the position estimate of the main body (since there is no direct position feedback), this filter has also a couple of advantages. It does not require the co-estimation of the foot location and has therefore a smaller filter state and reduced computational costs. Furthermore, by modeling the measurement error on the velocity level, outliers occurring due to slippage can be more easily

detected by a Mahalanobis based outlier detection. For more details on this filter please refer to [11].

The above approaches can also be adapted to legged robots with a *flat foot* model by extending the innovation term in eq. (6.58) to include the rotational constraint of the *flat foot* (eq. (6.9)) [117]. For a given foot orientation $\boldsymbol{q}_{F_i I}$, the innovation term can be formulated as

$$\boldsymbol{y}_{r,i}(\boldsymbol{x}, \tilde{\boldsymbol{\alpha}}) = \exp(\boldsymbol{n}_{cr,i}) \otimes \boldsymbol{q}_{F_i B}(\tilde{\boldsymbol{\alpha}}) \otimes \boldsymbol{q}_{BI} \otimes \boldsymbol{q}_{F_i I}^{-1}. \tag{6.61}$$

This innovation term comes in addition to the position term of eq. (6.58) and thus increases the amount of information that can be extracted from a ground contact. Consequently the observability of the system is improved, which basically means that less motion or ground contacts are required to make the system observable.

In general there are many options when designing a Kalman filter for legged state estimation. A large invariant is the use of an IMU-based process model as described in this section (eqs. (6.52) to (6.56)). This can be found in many different contributions [23, 34, 35, 91, 92, 113]. Also the work of Lin et al. [87] (presenting one of the first IMU-kinematics fusion methods) can be counted to this group since their update based IMU integration is statistically similar. When it comes to integrating kinematic information, however, methodologies diverge. For instance Reinstein and Hoffmann [113] propose to combine the EKF with a machine learning methodology. This processes pressure sensors and joint encoder measurements into an odometry output before merging it into the Kalman filter.

Different approaches can be applied if the robot is equipped with a high-grade IMU. In this case an accurate and reliable attitude estimation is often directly provided by the IMU. This strongly simplifies the problem and allows the use of direct matching methods in order to obtain position and velocity updates from kinematics (see section 4.2). Since this often yields noisy position and velocity estimates due to encoder noise and forward kinematics inaccuracies, they are sometimes fused with the IMU outputs. Because IMU measurements are used twice in that case, often without considering the cross-correlation of the involved term, this is statistically not the most sound approach. However, it represent a simple and reliable approach which provides accurate results (provided the robot is equipped with a high-accuracy IMU). Such methods are often applied on modern humanoids such as Boston Dynamics' Atlas robot. For example, Johnson et al. [72] linearly combine the kinematic position and velocity estimates with an IMU-driven model, by implementing a low pass filter. Others, including Fallon et al. [35] or Ma et al. [92], directly employ synthesized position/velocity feedback as update measurement in an IMU-driven EKF.

Inclusion of Exteroceptive Sensor Data

Using proprioceptive sensors only is often sufficient for providing a robot with a local ego-motion estimation including the attitude, velocity and incremental motion. However if a robot has to localize itself with respect to a world fixed reference frame, exteroceptive sensing becomes indispensable such that the robot is able to perceive its surrounding and infer its location within it. The most commonly used sensor modalities in this context are cameras [6, 23, 34, 91, 123] and laser range sensors [35, 124].

A common way of integrating exteroceptive sensory information is to process the data into a more compact form and, subsequently, to fuse it with the rest of the state estimation. For Kalman filter based estimation algorithms, the additional information can be fused into the estimation process by extending the update model. For instance, if the exteroceptive data can be processed to generate synthetic pose measurements of the main body $({}_I\tilde{\boldsymbol{r}}_{IB}, \tilde{\boldsymbol{q}}_{BI})$ a corresponding Kalman innovation term could be formulated as

$$\boldsymbol{y}_e = \begin{pmatrix} {}_I\tilde{\boldsymbol{r}}_{IB} - {}_I\boldsymbol{r}_{IB} + \boldsymbol{n}_r \\ \exp(\boldsymbol{n}_q) \otimes \tilde{\boldsymbol{q}}_{BI} \otimes \boldsymbol{q}_{BI}^{-1} \end{pmatrix}, \tag{6.62}$$

with $\boldsymbol{n}_r$ and $\boldsymbol{n}_q$ being additive Gaussian noise. Often, this approach has to be adapted to the actual setup. Typically, the inertial frame I or the body frame B of the pose measurements do not match with the Kalman filter internal inertial and body coordinate frames. In this case, the method has to be changed in order to account for a fixed offset between the coordinate frames. This can be done by offline calibration of the coordinate frames or by attempting to co-estimate the coordinate frame offset in the Kalman filter.

Chilian et al. [23] propose the use of a cloning approach in order to integrate the incremental pose measurements $({}_I\tilde{\boldsymbol{r}}_{IB_k}, \tilde{\boldsymbol{q}}_{B_kB_{k-1}})$ from a stereo vision odometry algorithm. The cloning approach augments the filter state in order to include the last pose $({}_I\boldsymbol{r}_{IB_{k-1}}, \boldsymbol{q}_{B_{k-1}I})$. This is required for remaining statistically consistent when formulating the incremental pose update which relates the current pose estimate to the last pose estimate:

$$\boldsymbol{y}_{ei,k} = \begin{pmatrix} {}_I\tilde{\boldsymbol{r}}_{IB_k} - ({}_I\boldsymbol{r}_{IB_{k-1}} + \boldsymbol{C}(\boldsymbol{q}_{B_{k-1}I})^T {}_{B_{k-1}}\boldsymbol{r}_{B_{k-1}B_k}) + \boldsymbol{n}_r \\ \exp(\boldsymbol{n}_q) \otimes \tilde{\boldsymbol{q}}_{B_kB_{k-1}} \otimes \boldsymbol{q}_{B_{k-1}I} \otimes \boldsymbol{q}_{B_kI}^{-1} \end{pmatrix}. \tag{6.63}$$

Neglecting this would quickly lead to inconsistencies due to ignored cross-correlations between the synthetic measurements and the filter state.

The use of laser range data for state estimation is less often observed. This is due to the more costly deployment of the sensor as well as the more involved processing of the data, which is even further complicated if the laser sensor is in motion during scanning. In order to overcome the issue of a moving sensor frame, Fallon et al. [35] use a particle filter to match a 2D scan of the environment to a prior map and thereby generate an update measurement for their IMU-driven EKF.

4.4 Exploitation of Dynamic Models and Measurements

Many legged robots are equipped with sensors that can measure joint or contact forces. In combination with an appropriate dynamic motion model (see eq. (6.1)), these measurements provide a further source of information for ego-motion estimation. Joint and contact forces can be directly related to the second order derivatives of position and attitude quantities. However, if no additional measurements or assumptions are provided, double integration of this noisy measurements will inevitably lead to drift of the estimated pose.

A common issue when integrating dynamic quantities are the considerable model inaccuracies. It is often difficult to precisely determine all model parameters such as the center of mass or moment of inertia. Furthermore, numerical derivatives of quantities such as joint encoder are involved in most relations and amplify the overall uncertainty of the employed dynamic models. In certain scenarios this motivates the introduction of model simplifications or approximations as the additional modeling inaccuracies become negligible.

Motion Estimation from Joint Torque Measurements

A possible approach for leveraging joint or contact force measurements will be discussed in the following. The multi-body equation of motion for a legged robot with n degrees of freedom and m independent contact constraints as described in eq. (6.1) and eq. (6.3) is

$$\boldsymbol{M}(\boldsymbol{\theta})\dot{\boldsymbol{u}} + \boldsymbol{b}(\boldsymbol{\theta}, \boldsymbol{u}) + \boldsymbol{g}(\boldsymbol{\theta}) + \boldsymbol{J}_c^T(\boldsymbol{\theta})\boldsymbol{F}_c = \boldsymbol{S}^T\boldsymbol{\tau}, \tag{6.64}$$

$$\dot{\boldsymbol{J}}_c(\boldsymbol{\theta})\boldsymbol{u} + \boldsymbol{J}_c(\boldsymbol{\theta})\dot{\boldsymbol{u}} = \boldsymbol{0}. \tag{6.65}$$

These equations can be solved for the generalized accelerations $\dot{\boldsymbol{u}}$, which can drive the process model of an EKF. However, it must be ensured that the entire equation only depends on the filter state and measured quantities. If, for instance, joint positions $\tilde{\boldsymbol{\alpha}}$ and joint forces $\tilde{\boldsymbol{\tau}}$ are measured, the contact forces $\boldsymbol{F}_c$ remain as unknown quantities. This can be overcome by left-multiplying the first part of the equation of motion by the right-null-space matrix $\boldsymbol{N}_c(\boldsymbol{\theta})$ of the contact Jacobian (satisfying $\boldsymbol{J}_c(\boldsymbol{\theta})\boldsymbol{N}_c(\boldsymbol{\theta}) = \boldsymbol{0}$):

$$\boldsymbol{N}_c^T(\boldsymbol{\theta})\boldsymbol{M}(\boldsymbol{\theta})\dot{\boldsymbol{u}} + \boldsymbol{N}_c^T(\boldsymbol{\theta})\boldsymbol{b}(\boldsymbol{\theta}, \boldsymbol{u}) + \boldsymbol{N}_c^T(\boldsymbol{\theta})\boldsymbol{g}(\boldsymbol{\theta}) = \boldsymbol{N}_c^T(\boldsymbol{\theta})\boldsymbol{S}^T\boldsymbol{\tau}. \tag{6.66}$$

For a fully-actuated system, the upper equation looses dimensions equals to the number of contact constraint m. The remaining n scalar equations can consequently be solved for the generalized accelerations:

$$\dot{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{N}_c^T(\boldsymbol{\theta})\boldsymbol{M}(\boldsymbol{\theta}) \\ \boldsymbol{J}_c(\boldsymbol{\theta}) \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{N}_c^T(\boldsymbol{\theta})\boldsymbol{S}^T\boldsymbol{\tau} - \boldsymbol{N}_c^T(\boldsymbol{\theta})\boldsymbol{b}(\boldsymbol{\theta}, \boldsymbol{u}) - \boldsymbol{N}_c^T(\boldsymbol{\theta})\boldsymbol{g}(\boldsymbol{\theta}) \\ -\dot{\boldsymbol{J}}_c(\boldsymbol{\theta})\boldsymbol{u} \end{bmatrix}. \tag{6.67}$$

An EKF can now be set up where the filter state is composed of $\boldsymbol{x} = (\boldsymbol{\theta}, \boldsymbol{u})$ and where the process model is a double integration of the estimated generalized accelerations $\dot{\boldsymbol{u}}$. The filter update step includes the identity update model $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha} + \boldsymbol{n}_\alpha$ and potentially additional measurement updates related to further sensors.

While approaches along the lines of the presented example seem nice in theory, they have only rarely been implemented in practice. One issue is that the computation of the matrices involved in the equation of motion is expensive for more complex legged systems. Furthermore, the EKF requires the evaluation of the Jacobian of eq. (6.67) with respect to $\boldsymbol{\theta}$ and $\boldsymbol{u}$ which can be quite involving. Computing this for a complex model at high update rate is very close at the computational limits of employed hardware and, due to the rather limited benefits, has consequently only rarely been implemented. Recently, more efficient methods have been tested for computing derivatives of the equation of motion [103], but it remains to see how well they can be integrated into a state estimation framework.

One way to make the above problem computationally tractable is to avoid a recursive filtering approach and to solve for the best current estimate given that the last estimate is correct (i.e. ignoring estimation error of the last time step). This has been proposed by Xinjilefu et al. [143] where a quadratic programming optimization method minimizes the error in eq. (6.64) and integrates various sensor models (joint velocity, joint torques, IMU, contact constraints, contact forces) while being able to simultaneously consider inequality constraints. Dynamic quantities like mass matrix or Coriolis terms are formulated as terms depending on the last step only and the corresponding Jacobians are therefore not required. This strongly reduces the computational load and allows the estimation process to be run real-time at high frequency (e.g. 500 Hz).

Simplified Models

In order to avoid the full equation of motion, the dynamics of a legged robot can also be simplified and approximated by a Spring Loaded Inverted Pendulum (SLIP) model [53] or by a Linear Inverted Pendulum Model (LIPM) [129, 142]. These reduced order motion models are able to capture the most important dynamic characteristics of the complex system and can be directly embedded in a Kalman filter. While this results in stable and reliable estimators (as long as the actual system dynamics are close to the selected model), it often suffers from insufficient accuracy. Another alternative to the full 3D dynamics is to make use of 2D planar dynamics (in the sagittal plane) as proposed by Aoustin et al. [2], Lebastard et al. [82].

Quasistatic Inclination Estimation

A last approach which should be mentioned is the synthesis of inclination measurements based on a dynamic model. The robot must constantly counteract the effect of gravity. By measuring this effort, it is possible to estimate the direction of gravity. For low-speed motions a quasi-static assumption $\dot{\boldsymbol{u}} = \boldsymbol{u} = \boldsymbol{0}$ can be made [51] such that eq. (6.64) simplifies to

$$\boldsymbol{g}(\boldsymbol{\theta}) + \boldsymbol{J}_c^T(\boldsymbol{\theta})\boldsymbol{F}_c = \boldsymbol{S}^T\boldsymbol{\tau}. \tag{6.68}$$

If the joint force $\boldsymbol{\tau}$ and the joint angles $\boldsymbol{\alpha}$ are measured, this equation can be used to solve for the gravity direction (contained in $\boldsymbol{\theta}$). Moreover, if the system is fully-actuated even the contact forces $\boldsymbol{F}_c$ can be estimated.

5 Future directions and open problems

State-of-the-art ego-motion estimation algorithms have demonstrated to perform very well for regular operation. In particular, the continuously ongoing improvement of IMUs, which are becoming a standard sensor installed on every robot, largely contributes to reliable state estimation.

However, state estimation algorithms still need improvement for operation in more complex scenarios where the contact situation is unclear, the robot is slipping, or the terrain is unstable. In those cases, the robot has to detect that its kinematic constraint

with the surrounding are uncertain (similarly to what human are doing) and consequently rely more on other sensor modalities. Potential performance improvement can also be expected by a better integration and exploitation of the system dynamics. In this context, a combination of force measurements and IMU measurements could provide the state estimation with a redundant and consequently robust motion prior.

6 Handling 3D Rotations

A more detailed discussion on the following elaborations can be found in [16]. As members of the special orthogonal group $SO(3)$, 3D rotations possess a multiplication operation (which is not commutative), but unfortunately do not have a concept of addition. Consequently the subtraction and differentiation, which are essential for most sensor fusion algorithms, do not exist either. In order to overcome this issue the region around a specific rotation can be mapped to a 3D vector space (the so-called Lie algebra). This is often done by means of the exponential and logarithmic map at identity. There are different ways of selecting these maps and a common choice is:

$$\begin{aligned} \log :& SO(3) \to \mathbb{R}^3 && (6.69)\\ & \boldsymbol{q}_{\mathcal{BI}} \mapsto \log(\boldsymbol{q}_{\mathcal{BI}}) = \boldsymbol{\varphi}_{\mathcal{BI}}, \\ \exp :& \mathbb{R}^3 \to SO(3) && (6.70)\\ & \boldsymbol{\varphi}_{\mathcal{BI}} \mapsto \exp(\boldsymbol{\varphi}_{\mathcal{BI}}) = \boldsymbol{q}_{\mathcal{BI}}, && (6.71) \end{aligned}$$

where $\boldsymbol{\varphi}_{\mathcal{BI}}$ is the passive rotation vector of the rotation parametrized by the unit quaternion $\boldsymbol{q}_{\mathcal{BI}}$.

These maps can now be used to define a boxplus operator and a boxminus operator as follows:

$$\begin{aligned} \boxplus :& SO(3) \times \mathbb{R}^3 \to SO(3) && (6.72)\\ & \boldsymbol{q}, \boldsymbol{\varphi} \mapsto \exp(\boldsymbol{\varphi}) \otimes \boldsymbol{q}, \\ \boxminus :& SO(3) \times SO(3) \to \mathbb{R}^3 && (6.73)\\ & \boldsymbol{q}, \boldsymbol{p} \mapsto \log(\boldsymbol{q} \otimes \boldsymbol{p}^{-1}). \end{aligned}$$

They represent a local concept of addition and subtraction and fulfill the axioms required by Hertzberg et al. [59]:

$$\begin{aligned} \boldsymbol{q} \boxplus \boldsymbol{0} &= \boldsymbol{q}, && (6.74)\\ (\boldsymbol{q} \boxplus \boldsymbol{\varphi}) \boxminus \boldsymbol{q} &= \boldsymbol{\varphi}, && (6.75)\\ \boldsymbol{q} \boxplus (\boldsymbol{p} \boxminus \boldsymbol{q}) &= \boldsymbol{p}. && (6.76) \end{aligned}$$

The regular addition and subtraction in the definition of differentials can now be replaced by the boxplus and boxminus operators in order to compute derivatives of terms involving 3D rotations. This yields the following set of derivatives for commonly

encountered terms (refer to [16] for a derivation):

$$\partial/\partial t\left(\boldsymbol{q}_{\mathcal{BI}}(t)\right) = -{}_{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{IB}}(t), \tag{6.77}$$

$$\partial/\partial \boldsymbol{q}\left(\boldsymbol{C}(\boldsymbol{q})\boldsymbol{r}\right) = -\left(\boldsymbol{C}(\boldsymbol{q})\boldsymbol{r}\right)^{\times}, \tag{6.78}$$

$$\partial/\partial \boldsymbol{q}\left(\boldsymbol{q}^{-1}\right) = -\boldsymbol{C}(\boldsymbol{q})^{T}, \tag{6.79}$$

$$\partial/\partial \boldsymbol{q}\left(\boldsymbol{q}\otimes\boldsymbol{p}\right) = \boldsymbol{I}, \tag{6.80}$$

$$\partial/\partial \boldsymbol{p}\left(\boldsymbol{q}\otimes\boldsymbol{p}\right) = \boldsymbol{C}(\boldsymbol{q}), \tag{6.81}$$

$$\partial/\partial \boldsymbol{\varphi}\left(\exp(\boldsymbol{\varphi})\right) =: \boldsymbol{\Gamma}(\boldsymbol{\varphi}), \tag{6.82}$$

$$\partial/\partial \boldsymbol{q}\left(\log(\boldsymbol{q})\right) = \boldsymbol{\Gamma}^{-1}(\log(\boldsymbol{q})). \tag{6.83}$$

Please note that these terms can vary based on the selected convention. The analytical forms of the rotation matrix $\boldsymbol{C}(\boldsymbol{\varphi})$ and of the exponential differential matrix $\boldsymbol{\Gamma}(\boldsymbol{\varphi})$ are given by:

$$\boldsymbol{C}(\boldsymbol{\varphi}) = \boldsymbol{I} + \frac{\sin(\|\boldsymbol{\varphi}\|)\boldsymbol{\varphi}^{\times}}{\|\boldsymbol{\varphi}\|} + \frac{(1-\cos(\|\boldsymbol{\varphi}\|))\boldsymbol{\varphi}^{\times^2}}{\|\boldsymbol{\varphi}\|^2}, \tag{6.84}$$

$$\boldsymbol{\Gamma}(\boldsymbol{\varphi}) = \boldsymbol{I} + \frac{(1-\cos(\|\boldsymbol{\varphi}\|))\boldsymbol{\varphi}^{\times}}{\|\boldsymbol{\varphi}\|^2} + \frac{(\|\boldsymbol{\varphi}\|-\sin(\|\boldsymbol{\varphi}\|))\boldsymbol{\varphi}^{\times^2}}{\|\boldsymbol{\varphi}\|^3}. \tag{6.85}$$

7 Solving the Least Squares Problem for Multiple Point Feet

The goal is to solve the nonlinear least squares problem of eq. (6.43) (Lagrangian form):

$$\min_{\boldsymbol{t},\boldsymbol{q},\lambda} \sum_{k} \bar{\boldsymbol{e}}_k^T(\boldsymbol{t},\boldsymbol{q})\bar{\boldsymbol{e}}_k(\boldsymbol{t},\boldsymbol{q}) + \lambda(\boldsymbol{q}^T\boldsymbol{q}-1), \tag{6.86}$$

with the error term from eq. (6.41)

$$\bar{\boldsymbol{e}}_k(\boldsymbol{t},\boldsymbol{q}) = \bar{\boldsymbol{t}}\otimes\boldsymbol{q} - \bar{\boldsymbol{a}}_k\otimes\boldsymbol{q} + \boldsymbol{q}\otimes\bar{\boldsymbol{b}}_k, \tag{6.87}$$

and where $\boldsymbol{t}$ is a 3D vector and $\boldsymbol{q}$ is a unit quaternion representing the incremental translation and rotation, respectively. Setting the derivatives with respect to $\boldsymbol{t}$, $\boldsymbol{q}$ and λ to zero results in the following set of equations:

$$\sum_{k} \bar{\boldsymbol{e}}_k^T(\boldsymbol{t},\boldsymbol{q})\boldsymbol{R}(\boldsymbol{q})\boldsymbol{S}^T = \boldsymbol{0}, \tag{6.88}$$

$$\sum_{k} \bar{\boldsymbol{e}}_k^T(\boldsymbol{t},\boldsymbol{q})\left(\boldsymbol{L}(\bar{\boldsymbol{t}}) - \boldsymbol{L}(\bar{\boldsymbol{a}}_k) + \boldsymbol{R}(\bar{\boldsymbol{b}}_k)\right) + \lambda\boldsymbol{q}^T = \boldsymbol{0}, \tag{6.89}$$

$$\boldsymbol{q}^T\boldsymbol{q} - 1 = 0. \tag{6.90}$$

Expanding and transposing the first equations gives

$$\sum_{k} \boldsymbol{S}\boldsymbol{R}(\boldsymbol{q}^{-1})\left(\bar{\boldsymbol{t}}\otimes\boldsymbol{q} - \bar{\boldsymbol{a}}_k\otimes\boldsymbol{q} + \boldsymbol{q}\otimes\bar{\boldsymbol{b}}_k\right) = \boldsymbol{0}, \tag{6.91}$$

which can be transformed and simplified to

$$N\boldsymbol{t} - \sum_k \boldsymbol{a}_k + \sum_k \boldsymbol{C}(\boldsymbol{q})\boldsymbol{b}_k = \boldsymbol{0}, \tag{6.92}$$

where N is the number of *point feet* in contact with the ground. This can finally be rearranged to

$$\boldsymbol{t} = \boldsymbol{a} - \boldsymbol{C}(\boldsymbol{q})\boldsymbol{b}, \quad \boldsymbol{a} = \frac{\sum_k \boldsymbol{a}_k}{N}, \quad \boldsymbol{b} = \frac{\sum_k \boldsymbol{b}_k}{N}. \tag{6.93}$$

This means that the translation is obtained from the vector that maps the mean of the rotated $\boldsymbol{b}_k$ to the mean of the $\boldsymbol{a}_k$.

The quaternion form of $\boldsymbol{t}$ is

$$\bar{\boldsymbol{t}} = \bar{\boldsymbol{a}} - \boldsymbol{q} \otimes \bar{\boldsymbol{b}} \otimes \boldsymbol{q}^{-1}, \tag{6.94}$$

which can be inserted into the transposed form of eq. (6.89):

$$\sum_k \left(\boldsymbol{L}(\bar{\boldsymbol{a}} - \boldsymbol{q} \otimes \bar{\boldsymbol{b}} \otimes \boldsymbol{q}^{-1}) - \boldsymbol{L}(\bar{\boldsymbol{a}}_k) + \boldsymbol{R}(\bar{\boldsymbol{b}}_k)\right)^T \left(\boldsymbol{L}(\bar{\boldsymbol{a}} - \boldsymbol{q} \otimes \bar{\boldsymbol{b}} \otimes \boldsymbol{q}^{-1}) - \boldsymbol{L}(\bar{\boldsymbol{a}}_k) + \boldsymbol{R}(\bar{\boldsymbol{b}}_k)\right) \boldsymbol{q} + \lambda \boldsymbol{q} = \boldsymbol{0}. \tag{6.95}$$

Furthermore, since

$$\boldsymbol{L}\left(\boldsymbol{q} \otimes \bar{\boldsymbol{b}} \otimes \boldsymbol{q}^{-1}\right) \boldsymbol{q} = \boldsymbol{R}\left(\bar{\boldsymbol{b}}\right) \boldsymbol{q}, \tag{6.96}$$

and the summation of the second factor in eq. (6.95) equals 0:

$$\sum_k \left(\boldsymbol{L}(\bar{\boldsymbol{a}}) - \boldsymbol{R}(\bar{\boldsymbol{b}}) - \boldsymbol{L}(\bar{\boldsymbol{a}}_k) + \boldsymbol{R}(\bar{\boldsymbol{b}}_k)\right) = \boldsymbol{0}, \tag{6.97}$$

eq. (6.95) can be transformed to:

$$-\sum_k \left(\boldsymbol{L}(\bar{\boldsymbol{a}}) - \boldsymbol{R}(\bar{\boldsymbol{b}}) - \boldsymbol{L}(\bar{\boldsymbol{a}}_k) + \boldsymbol{R}(\bar{\boldsymbol{b}}_k)\right) \left(\boldsymbol{L}(\bar{\boldsymbol{a}}) - \boldsymbol{R}(\bar{\boldsymbol{b}}) - \boldsymbol{L}(\bar{\boldsymbol{a}}_k) + \boldsymbol{R}(\bar{\boldsymbol{b}}_k)\right) \boldsymbol{q} + \lambda \boldsymbol{q} = \boldsymbol{0}. \tag{6.98}$$

This has the form of an Eigenvector problem for $\boldsymbol{q}$:

$$\boldsymbol{B}\boldsymbol{q} - \lambda \boldsymbol{q} = \boldsymbol{0}, \tag{6.99}$$

with

$$\boldsymbol{B} = \sum_k (\boldsymbol{A} - \boldsymbol{A}_k)(\boldsymbol{A} - \boldsymbol{A}_k) = \sum_k \boldsymbol{A}_k \boldsymbol{A}_k - \boldsymbol{A}\boldsymbol{A}, \tag{6.100}$$

$$\boldsymbol{A} = \sum_k \boldsymbol{A}_k, \quad \boldsymbol{A}_k = \boldsymbol{L}(\bar{\boldsymbol{a}}_k) - \boldsymbol{R}(\bar{\boldsymbol{b}}_k). \tag{6.101}$$

Paper III

State Estimation for Legged Robots – Consistent Fusion of Leg Kinematics and IMU

Michael Bloesch, Marco Hutter, Mark A. Hoepflinger, Stefan Leutenegger, Christian Gehring, C David Remy, Roland Siegwart

Abstract

This paper introduces a state estimation framework for legged robots that allows estimating the full pose of the robot without making any assumptions about the geometrical structure of its environment. This is achieved by means of an Observability Constrained Extended Kalman Filter that fuses kinematic encoder data with on-board IMU measurements. By including the absolute position of all footholds into the filter state, simple model equations can be formulated which accurately capture the uncertainties associated with the intermittent ground contacts. The resulting filter simultaneously estimates the position of all footholds and the pose of the main body. In the algorithmic formulation, special attention is paid to the consistency of the linearized filter: it maintains the same observability properties as the nonlinear system, which is a prerequisite for accurate state estimation. The presented approach is implemented in simulation and validated experimentally on an actual quadrupedal robot.

Published in:
Robotics Science and Systems Conference, 2012
DOI: 10.15607/RSS.2012.VIII.003

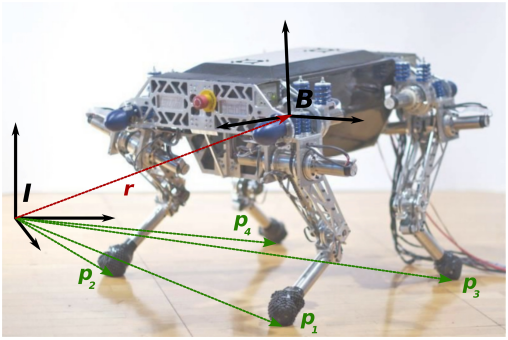

Figure 7.1: Experimental quadruped platform StarlETH [63]. The inertial and the body fixed coordinate frames $\boldsymbol{I}$ and $\boldsymbol{B}$ are depicted, as well as the absolute position vectors of the robot $\boldsymbol{r}$ and of the footholds $\boldsymbol{p_1}, \ldots, \boldsymbol{p_N}$. The presented EKF includes all foothold positions into the estimation process.

1 Introduction

Particularly in rough and highly unstructured environments in which we ultimately want to employ autonomous legged robots, postural controllers require fast and precise knowledge of the state of the robots they are regulating. Especially for dynamic locomotion, the underlying state estimation can quickly become a bottleneck in terms of achievable bandwidth, robustness, and locomotion speed. To achieve the required performance, a state estimator for legged robots should explicitly take into account that such systems are interacting with their environment via multiple *intermittent* ground contacts. Ignoring or neglecting the ground interaction will lead to computationally and sensory more "expensive" approaches, ranging from vision-based [23, 123, 134] to GPS-supported [27, 43] methods. In contrast to such approaches, we will show in the following that in cases where on-board sensors fully measure the internal kinematics of the robot as well as its inertial acceleration and rotational rate, precise information on the robot's pose can be made readily available.

One of the earliest approach exploiting information given by the leg kinematics was implemented by Lin et al. [86] in 2005 on a hexapod robot. Assuming that the robot is in contact with three of its six feet at all times and assuming completely flat terrain, they implemented a leg-based odometer. Their method requires the robots to follow a tripod gait and is affected by drift. In [87], the same group fused the leg-based odometer with data from an Inertial Measurement Unit (IMU) and thus is able to handle tripod running. Using the assumption of knowing the precise relief of the terrain, Chitta et al. [24] implemented a pose estimator based on a particle filter. It fuses leg kinematics and IMU in order to globally localize a robot.

Just very recently, three novel pose estimators have been presented that are all based on leg kinematics. Reinstein and Hoffmann [112] presented a data-driven approach using joint encoders, pressure sensors, and an on-board IMU. They searched

for significant sensory based indicators in order to determine the stride length when given a specific gait pattern. With this assumption, they successfully limited the position drift of the IMU and by appropriate training of the filter could additionally handle slippage. Chilian et al. [23] implemented a leg odometer based on point cloud matching for a hexapod robot, requiring a minimum of three feet in contact. It is based on a multisensor fusion algorithm that includes inertial measurements and visual odometry. Assuming planar spring-mass running, Gur and Saranli [53] proposed a generic, model-based state estimation technique.

In the presented approach we implement an Extended Kalman Filter (EKF) and choose an appropriate state vector in order to break down the estimation problem to the proper formulation of a few simple measurement equations. Without any assumption about the shape of the terrain, we are able to estimate the full state of the robot's main body, and we can provide an estimate of the ground geometry. By performing an observability analysis, we show that apart from the absolute position and yaw angle of the robot all other states can be precisely observed as long as at least one foot is in contact with the ground. This means that, after accumulating some drift during a flight phase, the pitch and roll angles become again fully observable when the robot regains ground contact and the corresponding estimation errors will decrease.

Only proprioceptive sensors are required and no assumptions are made concerning the type of gait or the number of robot legs. Little foot slippage and uncertainties on the leg kinematics can be handled as well. Due to current limitations of the control approach, dynamic gaits are currently evaluated in simulation only. Still, results obtained from static walking sequences on an actual quadrupedal platform (see Fig. 7.1) are presented and compared with ground truth measurements from an external motion tracker.

The structure of the paper is as follows. In section 2 a short overview of the sensory devices is provided. Subsequently, section 3 presents the design of an Extended Kalman Filter. Section 4 argues on the observability of the filter states and introduces observability constraints. Simulation and experimental validation are discussed in section 5.

2 Sensor Devices and Measurement Models

This section discusses the required sensors and the corresponding stochastic measurement models for a N legged robot. The particular model choices represent a trade-off between simplicity and accuracy. Throughout the paper, external disturbances and noise will be modeled as continuous white Gaussian noise or as discrete Gaussian noise processes. This is a coarse simplification, but can be handled by increasing the corresponding covariance matrix.

2.1 Forward Kinematics and Encoders

Incremental encoders provide access to the angular position of all joints. The corresponding encoder measurement vector $\tilde{\boldsymbol{\alpha}}$ of the joint angles vector $\boldsymbol{\alpha}$ is assumed to

be affected by discrete Gaussian noise $\boldsymbol{n}_\alpha$ with covariance matrix $\boldsymbol{R}_\alpha$:

$$\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha} + \boldsymbol{n}_\alpha. \tag{7.1}$$

Based on the known leg kinematics, the location of each foot can be computed with respect to the main body. However, due to erroneous calibration and possible errors in the kinematical model $\mathbf{lkin}_i(\cdot)$ of leg i, additive discrete Gaussian noise terms $\boldsymbol{n}_{s,i}$ are included in the model:

$$\boldsymbol{s}_i = \mathbf{lkin}_i(\boldsymbol{\alpha}) + \boldsymbol{n}_{s,i}, \tag{7.2}$$

where $\boldsymbol{s}_i$ represents the vector from the center of the main body to the contact point of leg i and where $\boldsymbol{R}_s$ is the covariance matrix of $\boldsymbol{n}_{s,i}$. Both, $\boldsymbol{s}_i$ and $\boldsymbol{n}_{s,i}$, are expressed in the body fixed frame $\boldsymbol{B}$.

2.2 Inertial Sensors

The IMU measures the proper acceleration $\boldsymbol{f}$ and the angular rate $\boldsymbol{\omega}$ of the robot's main body. The proper acceleration is related to the absolute acceleration $\boldsymbol{a}$ by

$$\boldsymbol{f} = \boldsymbol{C}(\boldsymbol{a} - \boldsymbol{g}), \tag{7.3}$$

where $\boldsymbol{C}$ is the matrix rotating coordinates of a vector expressed in the inertial coordinate frame $\boldsymbol{I}$ into the body coordinate frame $\boldsymbol{B}$. The IMU quantities $\boldsymbol{f}$ and $\boldsymbol{\omega}$ are assumed to be directly measured in the body coordinate frame $\boldsymbol{B}$. In order to describe the underlying stochastic process, the following continuous stochastic models are introduced:

$$\tilde{\boldsymbol{f}} = \boldsymbol{f} + \boldsymbol{b}_f + \boldsymbol{w}_f, \tag{7.4}$$

$$\dot{\boldsymbol{b}}_f = \boldsymbol{w}_{bf}, \tag{7.5}$$

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{b}_\omega + \boldsymbol{w}_\omega, \tag{7.6}$$

$$\dot{\boldsymbol{b}}_\omega = \boldsymbol{w}_{b\omega}. \tag{7.7}$$

The measured quantities $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{\omega}}$ are affected by additive white Gaussian noise processes $\boldsymbol{w}_f$ and $\boldsymbol{w}_\omega$ and by bias terms $\boldsymbol{b}_f$ and $\boldsymbol{b}_\omega$. The bias terms are modeled as Brownian motions and their derivatives can be represented by white Gaussian noise processes $\boldsymbol{w}_{bf}$ and $\boldsymbol{w}_{b\omega}$. The noise terms are specified by the corresponding covariance parameters $\boldsymbol{Q}_f$, $\boldsymbol{Q}_{bf}$, $\boldsymbol{Q}_\omega$, and $\boldsymbol{Q}_{b\omega}$. Following the paper of El-Sheimy et al. [32], they can be evaluated by examining the measured Allan variances. For the sake of simplicity each covariance parameter is assumed to be a diagonal matrix with identical diagonal entries.

3 State Estimation

As stated in the previous section, two different sources of data are available. Each of them provides information that can potentially contribute to the state estimate of the robot. In order to exploit this information an Extended Kalman Filter is designed. This section starts by defining the state vector of the filter and subsequently continues by formulating the filter models and equations.

3.1 Filter State Definition

The state vector of the filter has to be chosen such that the corresponding prediction and measurement equations can be stated in a clean and consistent manner. In this approach the state vector of the quadruped robot is composed of the position of the center of the main body $\boldsymbol{r}$, of the corresponding velocity $\boldsymbol{v}$ and of the quaternion $\boldsymbol{q}$ representing the rotation from the inertial coordinate frame $\boldsymbol{I}$ to the body coordinate frame $\boldsymbol{B}$. In order to consider the kinematics of the legs, the absolute positions of the N foot contact points $\boldsymbol{p}_i$ are included into the state vector. Together with the accelerometer bias $\boldsymbol{b}_f$ and the gyroscope bias $\boldsymbol{b}_\omega$ this yields the following state vector:

$$\boldsymbol{x} := \begin{pmatrix} \boldsymbol{r} & \boldsymbol{v} & \boldsymbol{q} & \boldsymbol{p}_1 & \cdots & \boldsymbol{p}_N & \boldsymbol{b}_f & \boldsymbol{b}_\omega \end{pmatrix}. \tag{7.8}$$

$\boldsymbol{r}$,$\boldsymbol{v}$ and all contact positions $\boldsymbol{p}_i$ are expressed in the inertial coordinate frame $\boldsymbol{I}$, whereas $\boldsymbol{b}_f$ and $\boldsymbol{b}_\omega$ are expressed in the body coordinate frame $\boldsymbol{B}$. Given a quaternion $\boldsymbol{q}$ the corresponding rotation matrix $\boldsymbol{C}$ can be easily determined.

The presented Extended Kalman Filter represents the uncertainties of the estimated state vector via the covariance matrix $\boldsymbol{P}$ of the corresponding state error vector $\boldsymbol{\delta x}$

$$\boldsymbol{P} := \mathrm{Cov}(\boldsymbol{\delta x}), \tag{7.9}$$

$$\boldsymbol{\delta x} := \begin{pmatrix} \boldsymbol{\delta r} & \boldsymbol{\delta v} & \boldsymbol{\delta \phi} & \boldsymbol{\delta p}_1 & \cdots & \boldsymbol{\delta p}_N & \boldsymbol{\delta b}_f & \boldsymbol{\delta b}_\omega \end{pmatrix}. \tag{7.10}$$

For the orientation state $\boldsymbol{q}$, special care has to be taken. It possesses 3 degrees of freedom and its covariance term should therefore also be represented by a 3 dimensional covariance matrix. Therefore the error of the pose is represented as a 3-dimensional rotation vector $\boldsymbol{\delta\phi}$. That is, if $\hat{\boldsymbol{q}}$ represents the estimate of the orientation quaternion, the error quaternion $\boldsymbol{\delta q}$ is defined by the relation

$$\boldsymbol{q} = \boldsymbol{\delta q} \otimes \hat{\boldsymbol{q}}, \tag{7.11}$$

where $\otimes$ is the quaternion multiplication operator and where the quaternion error is related to the error rotation vector by means of the map $\boldsymbol{\zeta}(\cdot)$:

$$\boldsymbol{\delta q} = \boldsymbol{\zeta}(\boldsymbol{\delta\phi}), \tag{7.12}$$

$$\boldsymbol{\zeta} : \boldsymbol{v} \mapsto \boldsymbol{\zeta}(\boldsymbol{v}) = \begin{bmatrix} \sin(\frac{1}{2}\|\boldsymbol{v}\|)\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} \\ \cos(\frac{1}{2}\|\boldsymbol{v}\|) \end{bmatrix}. \tag{7.13}$$

The inclusion of the foot contact positions into the filter state is the key point in the filter design, enabling a simple and consistent representation of the model equations. The leg kinematics measurements represent relative pose measurements between main body and foot contact, based on which the EKF is able to simultaneously correct the location of the foot contacts as well as the pose of the main body. In fact, the presented approach can be interpreted as a simultaneous localization and mapping (SLAM) algorithm, where the position of the actual foot contacts build up the map the robot is localized in.

3.2 Prediction model

The prediction equations are responsible for propagating the state from one timestep to the next. The IMU measurements $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{\omega}}$ are directly included here. Using (7.3)-(7.7), a set of continuous time differential equations can be formulated:

$$\dot{\boldsymbol{r}} = \boldsymbol{v}, \tag{7.14}$$

$$\dot{\boldsymbol{v}} = \boldsymbol{a} = \boldsymbol{C}^T(\tilde{\boldsymbol{f}} - \boldsymbol{b}_f - \boldsymbol{w}_f) + \boldsymbol{g}, \tag{7.15}$$

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})\boldsymbol{q} = \frac{1}{2}\boldsymbol{\Omega}(\tilde{\boldsymbol{\omega}} - \boldsymbol{b}_\omega - \boldsymbol{w}_\omega)\boldsymbol{q}, \tag{7.16}$$

$$\dot{\boldsymbol{p}}_i = \boldsymbol{C}^T\boldsymbol{w}_{p,i} \quad \forall i \in \{1, \ldots, N\}, \tag{7.17}$$

$$\dot{\boldsymbol{b}}_f = \boldsymbol{w}_{bf}, \tag{7.18}$$

$$\dot{\boldsymbol{b}}_\omega = \boldsymbol{w}_{b\omega}, \tag{7.19}$$

where $\boldsymbol{\Omega}(\cdot)$ maps an arbitrary rotational rate $\boldsymbol{\omega}$ to the 4x4 matrix used for representing the corresponding quaternion rate:

$$\boldsymbol{\Omega} : \boldsymbol{\omega} \mapsto \boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}. \tag{7.20}$$

While in principle the foot contacts are assumed to remain stationary, the white noise terms $\boldsymbol{w}_{p,i}$ in (7.17) with covariance parameter $\boldsymbol{Q}_{p,i}$ are added to the absolute foot positions in order to handle a certain amount of foot slippage. It is described in the body frame which allows tuning the magnitude of the noise terms in the different directions relative to the quadruped orientation (7.21). Furthermore, the noise parameter of a certain foothold is set to *infinity* (or to a very large value) whenever it has no ground contact. This enables the corresponding foothold to relocate and reset its position estimate when it regains ground contact, whereby the old foothold position is dropped from the estimation process. This is all that is required in order to handle intermittent contacts when stepping.

$$\boldsymbol{Q}_{p,i} = \begin{bmatrix} w_{p,i,x} & 0 & 0 \\ 0 & w_{p,i,y} & 0 \\ 0 & 0 & w_{p,i,z} \end{bmatrix}. \tag{7.21}$$

3.3 Measurement Model

Based on the kinematic model (7.2) a transformed measurement quantity is introduced for each leg i:

$$\tilde{\boldsymbol{s}}_i := \mathbf{lkin}_i(\tilde{\boldsymbol{\alpha}}) \tag{7.22}$$

$$\approx \mathbf{lkin}_i(\boldsymbol{\alpha}) + \boldsymbol{J}_{\mathrm{lkin},i}\boldsymbol{n}_\alpha \tag{7.23}$$

$$\approx \boldsymbol{s}_i \underbrace{-\boldsymbol{n}_{s,i} + \boldsymbol{J}_{\mathrm{lkin},i}\boldsymbol{n}_\alpha}_{\boldsymbol{n}_i}. \tag{7.24}$$

The linearized noise effect from the encoders (7.1) and the noise from the foothold position are joined into a new measurement noise quantity $\boldsymbol{n}_i$ with covariance matrix $\boldsymbol{R}_i$:

$$\boldsymbol{R}_i = \boldsymbol{R}_s + \boldsymbol{J}_{\text{lkin},i} \boldsymbol{R}_\alpha \boldsymbol{J}_{\text{lkin},i}^T, \tag{7.25}$$

where $\boldsymbol{J}_{\text{lkin},i}$ is the Jacobian of the kinematics of leg i with respect to the joint angles $\boldsymbol{\alpha}_i$ of the same leg:

$$\boldsymbol{J}_{\text{lkin},i} := \frac{\partial \mathbf{lkin}_i(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}_i} \qquad i \in \{1, \ldots, N\}. \tag{7.26}$$

$\tilde{\boldsymbol{s}}_i$ is the measurement of the position of the foot contact i with respect to the body coordinate frame $\boldsymbol{B}$ which can also be expressed as the absolute position of the foot contact minus the absolute position of the robot rotated into the body frame.

$$\tilde{\boldsymbol{s}}_i = \boldsymbol{C}(\boldsymbol{p}_i - \boldsymbol{r}) + \boldsymbol{n}_i. \tag{7.27}$$

3.4 Extended Kalman Filter Equations

For the subsequent linearization and discretization of the above models, the following auxiliary quantity is introduced:

$$\boldsymbol{\Gamma}_n := \sum_{i=0}^{\infty} \frac{\Delta t^{i+n}}{(i+n)!} \boldsymbol{\omega}^{\times i}, \tag{7.28}$$

where the $(\cdot)^\times$ superscript is used to represent the skew-symmetric matrix obtained from a vector. It draws on the series expansion of the matrix exponential. For $n = 0$ it yields:

$$\boldsymbol{\Gamma}_0 = \sum_{i=0}^{\infty} \frac{\left(\Delta t \boldsymbol{\omega}^\times\right)^i}{i!} = \exp\left(\Delta t \boldsymbol{\omega}^\times\right). \tag{7.29}$$

This means that $\boldsymbol{\Gamma}_0$ represents the incremental rotation matrix if rotating an arbitrary coordinate frame with a rotational rate of $-\boldsymbol{\omega}$ for Δt seconds. There exists a closed form expression for $\boldsymbol{\Gamma}_n$ that can be efficiently numerically evaluated (similar to Rodrigues' rotation formula).

Prediction Step

A standard filtering convention is employed: at time step k the *a priori* state estimate is represented by $\hat{\boldsymbol{x}}_k^-$, the *a posteriori* state estimate by $\hat{\boldsymbol{x}}_k^+$. Assuming zero-order hold for the measured quantities $\tilde{\boldsymbol{f}}_k$ and $\tilde{\boldsymbol{\omega}}_k$, and neglecting the effect of the incremental

rotation, equations (7.14)-(7.19) can be discretized to:

$$\hat{\boldsymbol{r}}_{k+1}^{-} = \hat{\boldsymbol{r}}_{k}^{+} + \Delta t \hat{\boldsymbol{v}}_{k}^{+} + \frac{\Delta t^2}{2}(\hat{\boldsymbol{C}}_{k}^{+T}\hat{\boldsymbol{f}}_k + \boldsymbol{g}), \tag{7.30}$$

$$\hat{\boldsymbol{v}}_{k+1}^{-} = \hat{\boldsymbol{v}}_{k}^{+} + \Delta t(\hat{\boldsymbol{C}}_{k}^{+T}\hat{\boldsymbol{f}}_k + \boldsymbol{g}), \tag{7.31}$$

$$\hat{\boldsymbol{q}}_{k+1}^{-} = \boldsymbol{\zeta}(\Delta t \hat{\boldsymbol{\omega}}_k) \otimes \hat{\boldsymbol{q}}_{k}^{+}, \tag{7.32}$$

$$\hat{\boldsymbol{p}}_{i,k+1}^{-} = \hat{\boldsymbol{p}}_{i,k}^{+} \quad \forall i \in \{1, \dots, N\}, \tag{7.33}$$

$$\hat{\boldsymbol{b}}_{f,k+1}^{-} = \hat{\boldsymbol{b}}_{f,k}^{+}, \tag{7.34}$$

$$\hat{\boldsymbol{b}}_{\omega,k+1}^{-} = \hat{\boldsymbol{b}}_{\omega,k}^{+}, \tag{7.35}$$

with the bias corrected IMU measurements

$$\hat{\boldsymbol{f}}_k = \tilde{\boldsymbol{f}}_k - \hat{\boldsymbol{b}}_{f,k}^{+}, \tag{7.36}$$

$$\hat{\boldsymbol{\omega}}_k = \tilde{\boldsymbol{\omega}}_k - \hat{\boldsymbol{b}}_{\omega,k}^{+}. \tag{7.37}$$

In order to correctly propagate the covariance matrix through the state dynamics, a set of linear differential equations describing the error dynamics is derived from (7.14)-(7.19) where all higher order terms were neglected:

$$\dot{\boldsymbol{\delta r}} = \boldsymbol{\delta v}, \tag{7.38}$$

$$\dot{\boldsymbol{\delta v}} = -\boldsymbol{C}^T \boldsymbol{f}^{\times} \boldsymbol{\delta \phi} - \boldsymbol{C}^T \boldsymbol{\delta b}_f - \boldsymbol{C}^T \boldsymbol{w}_f, \tag{7.39}$$

$$\dot{\boldsymbol{\delta \phi}} = -\boldsymbol{\omega}^{\times} \boldsymbol{\delta \phi} - \boldsymbol{\delta b}_\omega - \boldsymbol{w}_\omega, \tag{7.40}$$

$$\dot{\boldsymbol{\delta p}}_i = \boldsymbol{C}^T \boldsymbol{w}_{p,i} \quad \forall i \in \{1, \dots, N\}, \tag{7.41}$$

$$\dot{\boldsymbol{\delta b}}_f = \boldsymbol{w}_{bf}, \tag{7.42}$$

$$\dot{\boldsymbol{\delta b}}_\omega = \boldsymbol{w}_{b\omega}. \tag{7.43}$$

For the subsequent discretization, Van Loan's results [137] and the relation (7.28) can be applied to get the discrete linearized error dynamics matrix $\boldsymbol{F}_k$ and the discrete process noise covariance matrix $\boldsymbol{Q}_k$ (for readability only one foothold estimate is depicted):

$$\boldsymbol{F}_k = \begin{bmatrix} \mathbb{I} & \Delta t \mathbb{I} & -\frac{\Delta t^2}{2}\hat{\boldsymbol{C}}_{k}^{+T}\hat{\boldsymbol{f}}_k^{\times} & 0 & -\frac{\Delta t^2}{2}\hat{\boldsymbol{C}}_{k}^{+T} & 0 \\ 0 & \mathbb{I} & -\Delta t\hat{\boldsymbol{C}}_{k}^{+T}\hat{\boldsymbol{f}}_k^{\times} & 0 & -\Delta t\hat{\boldsymbol{C}}_{k}^{+T} & 0 \\ 0 & 0 & \hat{\boldsymbol{\Gamma}}_{0,k}^{T} & 0 & 0 & -\hat{\boldsymbol{\Gamma}}_{1,k}^{T} \\ 0 & 0 & 0 & \mathbb{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbb{I} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbb{I} \end{bmatrix}, \tag{7.44}$$

$$\begin{bmatrix} \frac{\Delta t^3}{3}\boldsymbol{Q}_f + \frac{\Delta t^5}{20}\boldsymbol{Q}_{bf} & \frac{\Delta t^2}{2}\boldsymbol{Q}_f + \frac{\Delta t^4}{8}\boldsymbol{Q}_{bf} & 0 \\ \frac{\Delta t^2}{2}\boldsymbol{Q}_f + \frac{\Delta t^4}{8}\boldsymbol{Q}_{bf} & \Delta t\boldsymbol{Q}_f + \frac{\Delta t^3}{3}\boldsymbol{Q}_{bf} & 0 \\ 0 & 0 & \Delta t\boldsymbol{Q}_\omega + (\hat{\boldsymbol{\Gamma}}_{3,k} + \hat{\boldsymbol{\Gamma}}_{3,k}^T)\boldsymbol{Q}_{b\omega} \\ 0 & 0 & 0 \\ -\frac{\Delta t^3}{6}\boldsymbol{Q}_{bf}\hat{\boldsymbol{C}}_k^+ & -\frac{\Delta t^2}{2}\boldsymbol{Q}_{bf}\hat{\boldsymbol{C}}_k^+ & 0 \\ 0 & 0 & -\boldsymbol{Q}_{b\omega}\hat{\boldsymbol{\Gamma}}_{2,k} \end{bmatrix}$$
$$\begin{bmatrix} 0 & -\frac{\Delta t^3}{6}\hat{\boldsymbol{C}}_k^{+T}\boldsymbol{Q}_{bf} & 0 \\ 0 & -\frac{\Delta t^2}{2}\hat{\boldsymbol{C}}_k^{+T}\boldsymbol{Q}_{bf} & 0 \\ 0 & 0 & -\hat{\boldsymbol{\Gamma}}_{2,k}^T\boldsymbol{Q}_{b\omega} \\ \Delta t\hat{\boldsymbol{C}}_k^{+T}\boldsymbol{Q}_p\hat{\boldsymbol{C}}_k^+ & 0 & 0 \\ 0 & \Delta t\boldsymbol{Q}_{bf} & 0 \\ 0 & 0 & \Delta t\boldsymbol{Q}_{b\omega} \end{bmatrix} = \boldsymbol{Q}_k.$$

By linearly combining two Gaussian distributions the Extended Kalman Filter stipulates the following *a priori* estimate of the covariance matrix at the timestep $k+1$:

$$\boldsymbol{P}_{k+1}^- = \boldsymbol{F}_k\boldsymbol{P}_k^+\boldsymbol{F}_k^T + \boldsymbol{Q}_k. \tag{7.45}$$

Update Step

The measurement residual, also called innovation, is the difference between actual measurements and their predicted value:

$$\boldsymbol{y}_k := \begin{pmatrix} \tilde{\boldsymbol{s}}_{1,k} - \hat{\boldsymbol{C}}_k^-(\hat{\boldsymbol{p}}_{1,k}^- - \hat{\boldsymbol{r}}_k^-) \\ \vdots \\ \tilde{\boldsymbol{s}}_{N,k} - \hat{\boldsymbol{C}}_k^-(\hat{\boldsymbol{p}}_{N,k}^- - \hat{\boldsymbol{r}}_k^-) \end{pmatrix}. \tag{7.46}$$

Considering the error states and again neglecting all higher order terms, it can be derived that the errors of the predicted leg kinematics measurements are given by:

$$\begin{aligned} \boldsymbol{s}_{i,k} - \hat{\boldsymbol{C}}_k^-(\hat{\boldsymbol{p}}_{i,k}^- - \hat{\boldsymbol{r}}_k^-) \approx &-\hat{\boldsymbol{C}}_k^-\boldsymbol{\delta r}_k^- + \hat{\boldsymbol{C}}_k^-\boldsymbol{\delta p}_{i,k}^- \\ &+ \left(\hat{\boldsymbol{C}}_k^-(\boldsymbol{p}_{i,k}^- - \boldsymbol{r}_k^-)\right)^\times \boldsymbol{\delta\phi}_k^-. \end{aligned} \tag{7.47}$$

With this the measurement Jacobian $\boldsymbol{H}_k$ can be evaluated:

$$\begin{aligned} \boldsymbol{H}_k &= \frac{\partial \boldsymbol{y}_k}{\partial \hat{\boldsymbol{x}}_k} \\ &= \begin{bmatrix} -\hat{\boldsymbol{C}}_k^- & 0 & \left(\hat{\boldsymbol{C}}_k^-(\hat{\boldsymbol{p}}_{1,k}^- - \hat{\boldsymbol{r}}_k^-)\right)^\times & \hat{\boldsymbol{C}}_k^- & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ -\hat{\boldsymbol{C}}_k^- & 0 & \left(\hat{\boldsymbol{C}}_k^-(\hat{\boldsymbol{p}}_{N,k}^- - \hat{\boldsymbol{r}}_k^-)\right)^\times & 0 & \cdots & \hat{\boldsymbol{C}}_k^- & 0 & 0 \end{bmatrix}. \end{aligned}$$

Stacking the single measurement noise matrices (7.25) returns the total measurement noise matrix:

$$\boldsymbol{R}_k = \begin{bmatrix} \boldsymbol{R}_{1,k} & & \\ & \ddots & \\ & & \boldsymbol{R}_{N,k} \end{bmatrix}. \tag{7.48}$$

Finally the *a priori* state estimate can be merged with the current measurements, where the Extended Kalman Filter states the following update equations:

$$\boldsymbol{S}_k := \boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^T + \boldsymbol{R}_k, \tag{7.49}$$

$$\boldsymbol{K}_k := \boldsymbol{P}_k^- \boldsymbol{H}_k^T \boldsymbol{S}_k^{-1}, \tag{7.50}$$

$$\boldsymbol{\Delta x}_k := \boldsymbol{K}_k \boldsymbol{y}_k, \tag{7.51}$$

$$\boldsymbol{P}_k^+ := (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_k^- \tag{7.52}$$

where $\boldsymbol{S}_k$ represents the innovation covariance, $\boldsymbol{K}_k$ the Kalman gain, $\boldsymbol{\Delta x}_k$ the resulting correction vector and $\boldsymbol{P}_k^+$ the *a posteriori* estimate of the state covariance matrix. Given $\boldsymbol{\Delta x}_k$ the state estimate can be updated. Again the orientation state requires special attention. Although the quaternion is of dimension 4, the extracted rotational correction $\boldsymbol{\Delta \phi}_k$ has only 3 dimensions. It basically represents the 3D rotation vector that needs to be applied to correct the predicted quaternion:

$$\hat{\boldsymbol{q}}_k^+ = \boldsymbol{\zeta}(\boldsymbol{\Delta \phi}_k) \otimes \hat{\boldsymbol{q}}_k^-. \tag{7.53}$$

4 Observability Analysis

4.1 Nonlinear Observability Analysis

Analyzing the observability characteristics of the underlying nonlinear system reveals the theoretical limitations of state estimation and can validate the employed approach. Based on the paper of Hermann and Krener [57] a nonlinear observability analysis is performed. In order to remain analytically tractable robocentric coordinates are introduced. The coordinate transformation is bijective and will thus not change the observability characteristics. Given the current operating point by

$$\boldsymbol{x}^* := \begin{pmatrix} \boldsymbol{r}^* & \boldsymbol{v}^* & \boldsymbol{q}^* & \boldsymbol{p}_1^* & \cdots & \boldsymbol{p}_N^* & \boldsymbol{b}_f^* & \boldsymbol{b}_\omega^* \end{pmatrix} \tag{7.54}$$

the following coordinate transformation is introduced:

$$\boldsymbol{z} := \begin{bmatrix} \boldsymbol{s}_1 \\ \vdots \\ \boldsymbol{s}_N \\ \bar{\boldsymbol{v}} \\ \bar{\boldsymbol{b}}_\omega \\ \bar{\boldsymbol{q}} \\ \bar{\boldsymbol{b}}_f \\ \bar{\boldsymbol{r}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}(\boldsymbol{p}_1 - \boldsymbol{r}) \\ \vdots \\ \boldsymbol{C}(\boldsymbol{p}_N - \boldsymbol{r}) \\ \boldsymbol{C}\boldsymbol{v} \\ \boldsymbol{b}_\omega - \boldsymbol{b}_\omega^* \\ \boldsymbol{q} \otimes \boldsymbol{q}^{*-1} \\ \boldsymbol{b}_f - \boldsymbol{b}_f^* \\ \boldsymbol{C}\boldsymbol{r} \end{bmatrix}. \tag{7.55}$$

The quantities in (7.55) are ordered such that a nice row echelon form results. The corresponding prediction model (7.14)-(7.19) and measurement equation (7.27) will be transformed to

$$\dot{\boldsymbol{z}} := \begin{bmatrix} (\boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega)^\times \boldsymbol{s}_1 - \bar{\boldsymbol{v}} \\ \vdots \\ (\boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega)^\times \boldsymbol{s}_N - \bar{\boldsymbol{v}} \\ (\boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega)^\times \bar{\boldsymbol{v}} + \boldsymbol{f} - \bar{\boldsymbol{b}}_f + \bar{\boldsymbol{C}}\boldsymbol{C}^* \boldsymbol{g} \\ 0 \\ \boldsymbol{\Omega}(\boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega)\bar{\boldsymbol{q}} \\ 0 \\ (\boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega)^\times \bar{\boldsymbol{r}} + \bar{\boldsymbol{v}} \end{bmatrix}, \tag{7.56}$$

$$\tilde{\boldsymbol{s}}_i = \boldsymbol{s}_i \qquad i \in \{1, \ldots, N\} \tag{7.57}$$

where all noise terms were disregarded as they have no influence on the observability and where $\bar{\boldsymbol{C}}$ and $\boldsymbol{C}^*$ represent the rotation matrices corresponding to $\bar{\boldsymbol{q}}$ and to $\boldsymbol{q}^*$, respectively.

The observability of the transformed system can now be analyzed. In contrast to the linear case, Lie-derivatives need to be computed in order to evaluate the observability matrix. By applying a few row-operations and by directly including the transformed operating point

$$\boldsymbol{z}^* := \begin{pmatrix} \boldsymbol{s}_1^* & \cdots & \boldsymbol{s}_N^* & \boldsymbol{C}^*\boldsymbol{v}^* & 0 & (0\ 0\ 0\ 1) & 0 & \boldsymbol{C}^*\boldsymbol{r}^* \end{pmatrix} \tag{7.58}$$

the observability matrix can be converted into a row echelon form. For the sake of readability the * are dropped again:

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} \mathbb{I} & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \mathbb{I} & 0 & 0 & 0 & 0 & 0 \\ 0 & \cdots & 0 & -\mathbb{I} & 0 & 0 & \boldsymbol{s}_1^\times & 0 \\ 0 & \cdots & 0 & 0 & \mathbb{I} & -2(\boldsymbol{C}\boldsymbol{g})^\times & \boldsymbol{\mathcal{O}}_1 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 2\boldsymbol{\omega}^\times(\boldsymbol{C}\boldsymbol{g})^\times & \boldsymbol{\mathcal{O}}_2 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & \boldsymbol{\Delta}\boldsymbol{s}_{i,j}^\times & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & \boldsymbol{\Delta}\boldsymbol{s}_{i,j}^\times \boldsymbol{\omega}^\times & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & \boldsymbol{\Delta}\boldsymbol{s}_{i,j}^\times \boldsymbol{\omega}^\times \boldsymbol{\omega}^\times & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & \boldsymbol{\mathcal{O}}_3 & 0 \\ & & & & & & \vdots & \\ 0 & \cdots & 0 & 0 & 0 & 0 & & 0 \end{bmatrix}, \tag{7.59}$$

$$\boldsymbol{\mathcal{O}}_1 = -\boldsymbol{s}_1^{\times}\boldsymbol{\omega}^{\times} - 2(\boldsymbol{C}\boldsymbol{v})^{\times}, \tag{7.60}$$

$$\begin{aligned}\boldsymbol{\mathcal{O}}_2 = &(\boldsymbol{s}_1^{\times}\boldsymbol{\omega}^{\times} + 3(\boldsymbol{C}\boldsymbol{v})^{\times})\boldsymbol{\omega}^{\times} - \boldsymbol{\omega}^{\times}(\boldsymbol{s}_1^{\times}\boldsymbol{\omega}^{\times} + 2(\boldsymbol{C}\boldsymbol{v})^{\times}) \\ &- (\boldsymbol{C}\boldsymbol{g})^{\times} - 2\boldsymbol{f}^{\times},\end{aligned} \tag{7.61}$$

$$\begin{aligned}\boldsymbol{\mathcal{O}}_3 = &\boldsymbol{\omega}^{\times}(\boldsymbol{s}_1^{\times}\boldsymbol{\omega}^{\times}\boldsymbol{\omega}^{\times} + 5(\boldsymbol{C}\boldsymbol{v})^{\times}\boldsymbol{\omega}^{\times} - 4\boldsymbol{f}^{\times} - 3(\boldsymbol{C}\boldsymbol{g})^{\times}) \\ &- (\boldsymbol{s}_1^{\times}\boldsymbol{\omega}^{\times}\boldsymbol{\omega}^{\times} + 4(\boldsymbol{C}\boldsymbol{v})^{\times}\boldsymbol{\omega}^{\times} - 3\boldsymbol{f}^{\times} - 2(\boldsymbol{C}\boldsymbol{g})^{\times})\boldsymbol{\omega}^{\times} \\ &- 4\boldsymbol{\omega}^{\times}(\boldsymbol{C}\boldsymbol{v})\boldsymbol{\omega}^{T},\end{aligned} \tag{7.62}$$

$$\boldsymbol{\Delta s}_{i,j} := \boldsymbol{s}_i - \boldsymbol{s}_j. \tag{7.63}$$

A full interpretation of this matrix is not within the scope of this paper. However, two essential points are emphasized. The four dimensional manifold composed of robot position and yaw angle (rotation around gravity vector $\boldsymbol{g}$) is *always* unobservable. This can be verified by looking at the tangential space spanned by the matrix

$$\bar{\boldsymbol{U}} = \begin{bmatrix} 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \mathbb{I} \\ 0 & \cdots & 0 & 0 & 0 & \frac{1}{2}(\boldsymbol{C}\boldsymbol{g})^T & 0 & 0 \end{bmatrix}^T, \tag{7.64}$$

$$0 = \boldsymbol{\mathcal{O}}\bar{\boldsymbol{U}}. \tag{7.65}$$

Infinitesimal errors $\boldsymbol{\Delta z} = \bar{\boldsymbol{U}}\boldsymbol{\epsilon}$ lying within the subspace of $\bar{\boldsymbol{U}}$ cannot be detected. Transforming this back to our original coordinates yields the tangential space

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{C}^T & 0 & 0 & \boldsymbol{C}^T & \cdots & \boldsymbol{C}^T & 0 & 0 \\ \boldsymbol{g}^T\boldsymbol{r}^{\times} & \boldsymbol{g}^T\boldsymbol{v}^{\times} & \boldsymbol{g}^T\boldsymbol{C}^T & \boldsymbol{g}^T\boldsymbol{p}_1^{\times} & \cdots & \boldsymbol{g}^T\boldsymbol{p}_N^{\times} & 0 & 0 \end{bmatrix}^T \tag{7.66}$$

where the upper row corresponds to a 3 dimensional translation of the inertial coordinate frame and where the lower row corresponds to a rotation of the inertial coordinate frame around the gravity axis $\boldsymbol{g}$.

The second point is that in some cases, the rank loss associated with the unobservable manifold can increase by up to 5 additional ranks. Table 7.1 depicts some of the cases. All cases which induce a rank loss require some singularities. It can thus be stated that statistically *almost surely* the unobservable space will be limited to absolute position and yaw angle (except for the case where there is no ground contact at all). Note that if the bias estimation is excluded, the unobservable subspace will be *invariantly of rank four*.

Unfortunately, typical operating points can lie very close to singular cases. The upper highlighted row in table 7.1 represents the case where the robot has at least 3 non co-linear ground contacts and where the rotation axis is not perpendicular to the gravity vector. The lower highlighted row represents the corresponding singular case where $\boldsymbol{\omega} = 0$ inducing a rank loss of 2. This proximity to singular cases can cause bad convergence quality. For this reason the filter is implemented in such a manner that the estimation of the accelerometer and gyroscope biases can be enabled or disabled at runtime. Thereby it is possible to disable the bias estimation for critical tasks. On the other hand special maneuvers can be derived from the conditions in table 7.1 which can properly estimate the bias states.

$\boldsymbol{\omega}$	$\boldsymbol{f}$	$\boldsymbol{v}$	$\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N$	Rank loss
$\boldsymbol{\omega} \cdot \boldsymbol{Cg} \neq 0$	*	*	not co-linear	0
$\boldsymbol{\omega} \cdot \boldsymbol{Cg} \neq 0$	$\det \boldsymbol{\mathcal{O}}_3 \neq 0$		at least one contact	0
$\boldsymbol{\omega} \cdot \boldsymbol{Cg} = 0$	*	*	at least one contact	≥ 1
0	*	*	at least one contact	≥ 2
0	*	*	not co-linear	2
0	0	*	$\boldsymbol{s}_1 = \ldots = \boldsymbol{s}_N$	3
0	$-1/2\boldsymbol{Cg}$	*	$\boldsymbol{s}_1 = \ldots = \boldsymbol{s}_N$	5

Table 7.1: Estimation scenarios and corresponding rank loss.

4.2 Observability Analysis of the Extended Kalman Filter

The filter makes use of a linearized and discretized version of the nonlinear system model:

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k \boldsymbol{x}_k + \boldsymbol{w}_{lin,k}, \tag{7.67}$$

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{n}_{lin,k}, \tag{7.68}$$

where errors caused by linearization or discretization are incorporated in the noise terms $\boldsymbol{w}_{lin,k}$ and $\boldsymbol{n}_{lin,k}$. The corresponding observability analysis will be performed by applying the concept of local observability matrices [22]: similar to the time-invariant case the observable subspace can be derived by analyzing the subspace spanned by the rows of a local observability matrix:

$$\boldsymbol{\mathcal{M}} = \begin{bmatrix} \boldsymbol{H}_k \\ \boldsymbol{H}_{k+1}\boldsymbol{F}_k \\ \boldsymbol{H}_{k+2}\boldsymbol{F}_{k+1}\boldsymbol{F}_k \\ \boldsymbol{H}_{k+3}\boldsymbol{F}_{k+2}\boldsymbol{F}_{k+1}\boldsymbol{F}_k \\ \vdots \end{bmatrix}. \tag{7.69}$$

The observability characteristics of the discrete linear time-varying system (7.67)-(7.68) can differ from those of the underlying nonlinear system (7.14)-(7.19),(7.27). This discrepancy can be caused by linearization/discretization effects as well as by noise effects. The effect of noise becomes particularly evident when contemplating the observability characteristics of a corresponding noiseless (ideal) system. For the presented system the effect of noise renders the yaw angle observable by preventing the evaluation of the Jacobians $\boldsymbol{F}_k$ and $\boldsymbol{H}_k$ around the true state and thereby increasing the numerical rank of the local observability matrix $\boldsymbol{\mathcal{M}}$. The spurious appearance of new observable states is strongly objectionable as it results in overconfident state estimation. The magnitude of this inconsistency depends on the noise ratio, but in the long run, it will always deteriorate the state estimate.

The above phenomenon has been observed earlier in the context of EKF-SLAM [61, 75]. Huang et al. [62] introduced the Observability Constrained Extended Kalman Filter in order to tackle this issue. The approach in this paper goes much along their idea: the unobservable subspace of the nonlinear system (7.66) should also be

unobservable in the linearized and discretized system (7.67)-(7.68). Mathematically, this can be imposed by adding the following constraint:

$$\boldsymbol{\mathcal{M}}\ \boldsymbol{U} = 0. \tag{7.70}$$

In order to meet this constraint Huang et al. evaluate the Jacobians at special operating points: instead of using the actual state estimate they use slightly altered values.

The approach in this paper tackles the observability problem by exploiting the following observation: the noiseless case does meet the constraint (7.70) because it perfectly fulfills the prediction equations (7.30)-(7.35) and thus the appropriate terms are canceled out. For the presented filter it suffices if the following constraints are introduced (where a * denotes the states or measurements around which Jacobians are evaluated):

$$\boldsymbol{r}^*_{k+1} = \boldsymbol{r}^*_k + \Delta t \boldsymbol{v}^*_k + \frac{\Delta t^2}{2}(\boldsymbol{C}^{*T}_k \boldsymbol{f}^*_{k,1} + \boldsymbol{g}), \tag{7.71}$$

$$\boldsymbol{v}^*_{k+1} = \boldsymbol{v}^*_k + \Delta t(\boldsymbol{C}^{*T}_k \boldsymbol{f}^*_{k,2} + \boldsymbol{g}), \tag{7.72}$$

$$\boldsymbol{q}^*_{k+1} = \boldsymbol{\zeta}(\boldsymbol{\omega}^*_k) \otimes \boldsymbol{q}^*_k, \tag{7.73}$$

$$\boldsymbol{p}^*_{i,k+1} = \boldsymbol{p}^*_{i,k} \quad \forall i \in \{1, \ldots, N\}. \tag{7.74}$$

Both, filter state and IMU measurements, are allowed to differ from their actual estimated quantities. However, in order to keep the linearization errors small the linearization point should remain as close as possible to the estimated state. Thus, given the timestep l_i of the last touch-down event of foot i, the first-ever available estimate is chosen for the linearization:

$$\boldsymbol{r}^*_k = \boldsymbol{r}^-_k, \quad \boldsymbol{v}^*_k = \boldsymbol{v}^-_k, \quad \boldsymbol{q}^*_k = \boldsymbol{q}^-_k, \tag{7.75}$$

$$\boldsymbol{p}^*_{i,k} = \boldsymbol{p}^-_{i,l_i} \quad \forall i \in \{1, \ldots, N\}. \tag{7.76}$$

This is in analogy to the First-Estimates Jacobian EKF of Huang et al. [61]. But, in general, the prediction constraints (7.71)-(7.73) are still not met. For this reason the additional terms $\boldsymbol{f}^*_{k,1}$, $\boldsymbol{f}^*_{k,2}$ and $\boldsymbol{\omega}^*_k$ were introduced. Now, by choosing

$$\boldsymbol{f}^*_{k,1} = \boldsymbol{C}^{*T}_k \left(\frac{\boldsymbol{r}^*_{k+1} - \boldsymbol{r}^*_k - \Delta t \boldsymbol{v}^*_k}{0.5\Delta t^2} - \boldsymbol{g} \right), \tag{7.77}$$

$$\boldsymbol{f}^*_{k,2} = \boldsymbol{C}^{*T}_k \left(\frac{\boldsymbol{v}^*_{k+1} - \boldsymbol{v}^*_k}{\Delta t} - \boldsymbol{g} \right), \tag{7.78}$$

$$\boldsymbol{\omega}^*_k = \boldsymbol{\zeta}^{-1} \left(\boldsymbol{q}^*_{k+1} \otimes \boldsymbol{q}^{*-1}_k \right) \tag{7.79}$$

all constraints can be easily met. The above quantities represent the IMU measurements that would arise when considering two subsequent filter prediction states at timestep k and $k+1$. Because the acceleration related measurements can differ if evaluated based on the position prediction or on the velocity prediction, two terms were introduced. This permits to keep the computation of the linearization quantities

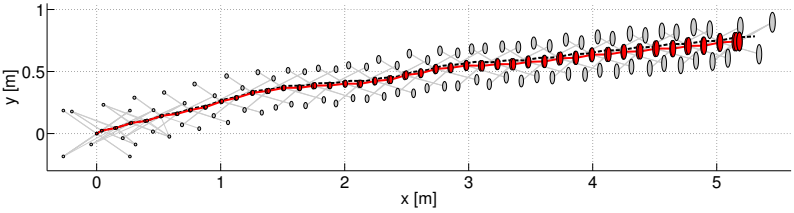

Figure 7.2: 2D view of a 5 m trotting sequence in simulation. Dashed line (in the background): ground-truth body trajectory. Darker ellipses (red): successive position estimates of the robot's main body. Light grey ellipses: estimates of the foothold positions. In both cases the ellipses are scaled depending on the corresponding standard deviation (1σ). The position error at the end amounts to less than 5% of the traveled distance.

simple and avoids complex optimization algorithms or oscillation provoking bindings between subsequent linearization points.

Computing the Jacobians $\boldsymbol{F}_k$ and $\boldsymbol{H}_k$ using the supplementary linearization quantities and evaluating the corresponding local observability matrix (7.69) yields:

$$\boldsymbol{\mathcal{M}} = \begin{bmatrix} -\mathbb{I}\ 0 & \boldsymbol{s}_{1,k}^{\times}\boldsymbol{C}_k^T & \mathbb{I} \cdots 0 & 0 & 0 \\ \vdots\ \vdots & \vdots & \vdots\ \ddots\ \vdots & \vdots & \vdots \\ -\mathbb{I}\ 0 & \boldsymbol{s}_{1,k}^{\times}\boldsymbol{C}_k^T & 0 \cdots \mathbb{I} & 0 & 0 \\ 0\ \ \mathbb{I} & (\boldsymbol{v}_k + \frac{\Delta t}{2}\boldsymbol{g})^{\times}\boldsymbol{C}_k^T & 0\ 0\ 0 & -\frac{\Delta t^2}{2}\boldsymbol{C}_k^T & \# \\ 0\ \ 0 & -\boldsymbol{g}^{\times} & 0\ 0\ 0 & \frac{1}{2}(\boldsymbol{C}_{k+1}^T + \boldsymbol{C}_k^T) & \# \\ 0\ \ 0 & 0 & 0\ 0\ 0 & \frac{1}{2}(\boldsymbol{C}_{k+2}^T - \boldsymbol{C}_k^T) & \# \\ 0\ \ 0 & 0 & 0\ 0\ 0 & \frac{1}{2}(\boldsymbol{C}_{k+3}^T - \boldsymbol{C}_{k+1}^T) & \# \\ & & & \vdots & \\ 0\ \ 0 & 0 & 0\ 0\ 0 & \vdots & \# \end{bmatrix}$$

whereby it is simple to test that the observability constraint (7.70) is satisfied. As a last side note: similarly to the nonlinear case, observability rank loss will again be induced when $\boldsymbol{\omega} \equiv 0$ and thus

$$\boldsymbol{C}_{k+2}^T - \boldsymbol{C}_k^T = 0. \tag{7.80}$$

5 Results and Discussion

Experiments are performed in simulation and on a real platform, whereby a series-elastic actuated quadruped is stabilized by a virtual model control approach [63] using the feedback of the pose estimator. The estimation of accelerometer and gyroscope biases is always enabled. In a first experiment the filter behavior is evaluated for a dynamic trotting gait within a simulation environment including realistic noise levels.

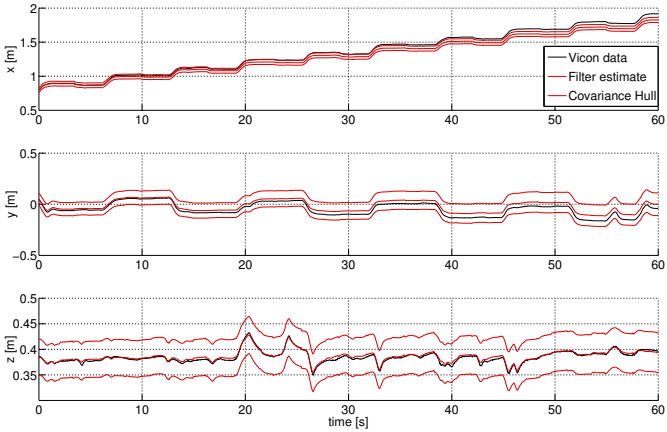

Figure 7.3: Comparison between estimated position and the motion capture system's position outputs. All three positions are affected by some drift, amounting up to 10 % of the traveled distance.

Fig. 7.2 shows results from a 15 s trajectory with a reference forward speed of 0.4 $^{\mathrm{m}}/_{\mathrm{s}}$. The uncertainties of the robot and of the foothold positions are represented by the corresponding 1σ-ellipses. The effects of unobservable absolute position and yaw angle can clearly be perceived. The leg kinematics measurements directly correlate the estimate of the main body position and the estimates of the foothold positions and thereby strongly limit the drift. Moreover, considering the correlations induced by the prediction model, the filter is able to properly correct the estimated quantities rendering the inclination angles and the velocities fully observable. Based on the resulting state estimate the quadruped can stabilize itself in a highly dynamic gait.

The second set of results is collected on a real platform. During the experiment independent ground truth measurements are provided by an external visual tracking system. A 60 s long static walking sequence where the robot moves approximately one meter forward is evaluated. By pushing and tilting the robot external disturbances are imposed on the slow locomotion pattern. Because the position is not fully observable, a slight drift occurs for the corresponding estimates (see Fig. 7.3), it can amount up to roughly 10 % of the traveled distance. Notable sources for the drift are the inaccurate leg kinematics and the fault-prone contact detection. The slightly longer actual robot shank explains the shorter estimated traveled distance (x direction). On the other hand, small perturbations are closely tracked by the filter. This is attested by very precise velocity estimates yielding RMS error values of less than 0.02 $^{\mathrm{m}}/_{\mathrm{s}}$ (see Fig. 7.4). Like the velocity states, the roll and pitch angles are fully observable as well and exhibit also very small estimation errors (see Fig. 7.5). The drift of the yaw angle is almost imperceivable. For all estimates the corresponding 3σ covariance-

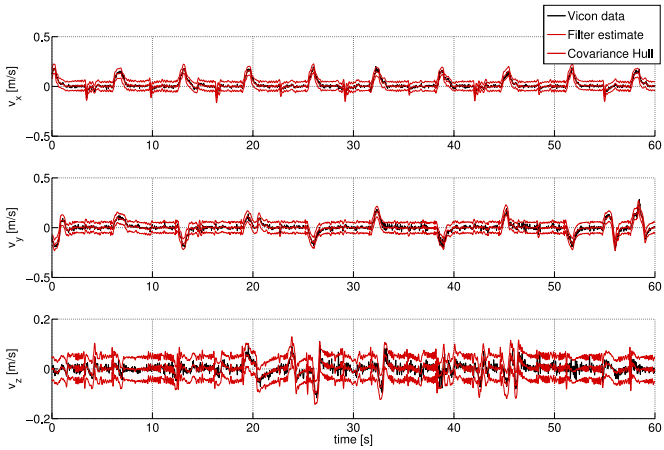

Figure 7.4: Comparison between estimated velocity and the motion capture system's numerical position derivatives. All three velocity estimates are fully observable and consequently can be tracked very accurately. The resulting RMS error values are 0.0111 m/s, 0.0153 m/s and 0.0126 m/s.

hull is plotted. Except for the x-position estimate, where model inaccuracies induce a significant offset, all estimate errors remain within the covariance-hull and thus validate the consistency of the presented approach.

6 Conclusion and Future Work

This paper presents a pose estimator for legged robots. It fuses information from leg kinematics and IMU data, whereby the model equations are kept simple and precise, and only a minimum of assumptions is introduced (mainly limited foot slippage). The filter can handle unknown terrain and arbitrary locomotion gaits. Through an observability analysis, it was shown that for non-degenerate cases only absolute position and yaw angle are not observable. Consequently, the roll and pitch angles as well as the robot's velocity can be accurately tracked, which was confirmed by the experimental results. Compared to proprioceptive sensor setups only, the obtained state estimate attains an unpreceded level of precision. The very generic formulation enables the filter to be extended with further sensory measurements and allows its implementation on various kinds of legged platforms.

Future work will include handling the unobservable states. Different approaches like introducing coordinate transformations, partitioning the unobservable manifold or implementing pseudo-measurements could be evaluated. Fusion with exteroceptive sensors will also be investigated. More aggressive locomotion needs to be further

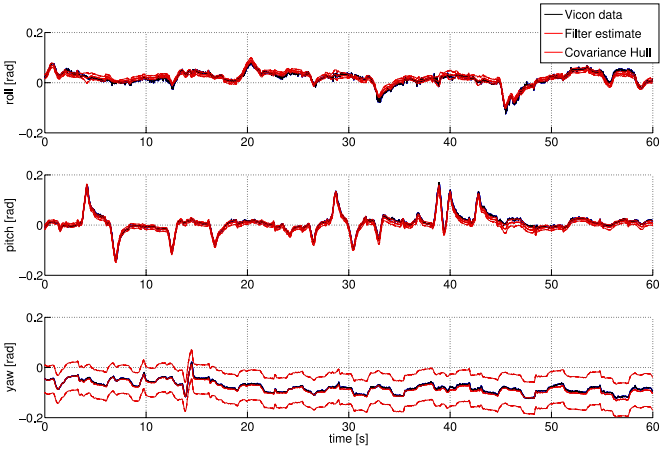

Figure 7.5: Comparison between estimated roll, pitch and yaw angle and the motion capture system's orientation outputs. Roll and pitch angle are fully observable and the filter produces very precise corresponding estimates, with angular error RMS of less than 0.5 deg (0.0088 rad and 0.0073 rad). The yaw angle drift is almost unnoticeable.

tested: while it has been validated in simulation, future work will include dynamic walking on the real quadruped platform.

Paper IV

State Estimation for Legged Robots on Unstable and Slippery Terrain

Michael Bloesch, Christian Gehring, Peter Fankhauser, Marco Hutter, Mark A. Hoepflinger, Roland Siegwart

Abstract

This paper presents a state estimation approach for legged robots based on stochastic filtering. The key idea is to extract information from the kinematic constraints given through the intermittent contacts with the ground and to fuse this information with inertial measurements. To this end, we design an unscented Kalman filter based on a consistent formulation of the underlying stochastic model. To increase the robustness of the filter, an outliers rejection methodology is included into the update step. Furthermore, we present the nonlinear observability analysis of the system, where, by considering the special nature of 3D rotations, we obtain a relatively simple form of the corresponding observability matrix. This yields, that, except for the global position and the yaw angle, all states are in general observable. This also holds if only one foot is in contact with the ground. The presented filter is evaluated on a real quadruped robot trotting over an uneven and slippery terrain.

Published in:
IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013
DOI: 10.1109/IROS.2013.6697236

1 Introduction

As the research in legged robotic design and control is resulting in increasingly performing platforms, the aspect of state estimation and perception of such machines becomes more and more important as well. In order to be able to leave structured and controlled lab environments and go into more uncertain, rough and difficult terrain, it is indispensable to endow legged robots with precise state estimation and perception capabilities. Consequently, different research groups explored the integration of perception devices on legged platforms [91, 123, 128]. In the present paper however, focus is set on the proper extraction of information contained in the kinematics of the robot and obtained from inertial sensors. While for most legged robots such data is readily available from on-board sensor devices, it also represents a very valuable source of high-bandwidth information for state estimation. In our opinion, the exploitation of this information is a prerequisite for fast and elaborate control of legged robots in unstructured and difficult environments and represents an important foundation for the inclusion of further sensor modalities like vision or LIDAR.

Roston et al. [116] presented one of the earliest navigation system which extracts information from leg kinematics. By matching the foot positions between two consecutive timesteps they compute the incremental motion of the main body. Further, they introduce a slip detection method which relies on the invariance of the distance between feet that are in contact with the ground. Several groups extend this idea, e.g., Gassmann et al. [43] introduce fuzzy weights, based on different sensor measurements, in order to describe how well a certain foot is in contact with the ground and fuse the resulting legged odometer with GPS data. Along similar lines Lin et al. [87] present a leg strain-based odometer and use an inertial measurement unit (IMU) for handling flight phases of their hexapod robot. Again based on contact point matching, Görner et al. [51] present a legged odometer where joint torques are used to estimate roll and pitch of a fully actuated hexapod. A common drawback of these methods is that the associated legged odometer requires at least 3 non-colinear feet in contact with the ground.

Other approaches range from data-driven methods to model based observers. For example, using joint encoders, pressure sensors, and IMU data, Reinstein and Hoffmann [113] search for significant sensory based indicators in order to determine stride length. While it requires training of the state estimation for new locomotion scenarios, it enables the handling of cases with significant foot slippage. Based on a two dimensional dynamic model, Lebastard et al. [82] designed a high-order sliding-mode observer for estimating the 2D posture of their bipedal robot during a walking gait. Assuming planar spring-mass running, Gur and Saranli [53] propose a generic, model-based state estimation technique. The major issues of these approaches are the requirement for a precise dynamic system model and the possible restriction to a specific type of motion.

The detection of outliers in the context of legged robotic state estimation has only scarcely been studied. Most approaches use some additional force sensing on the foot level and compare desired and actual forces in order to detect slippage [78]. More recently, Okita and Sommer [106] considered slip events being anomalies which can be detected by employing appropriate filtering methods. In a simplified 2D stick-slip experiment they showed how to detect slippage using smoothed innovation in

an Unscented Kalman filter (UKF) setup. Detecting anomalies or outliers in general filtering frameworks has been very widely analyzed. Ting et al. [133] as well as Agamennoni et al. [1] present outlier robust Kalman filtering by introducing more flexible noise models which allow the co-estimation of update noise parameters. Others investigated the use of non-Gaussian distributions which are less susceptible to outliers [127, 140].

The present paper is an extension to our prior work [10]. While following a similar overall approach, in the sense that accurate estimates of the full body pose are obtained by fusing information from an on-board IMU and kinematic measurements, the presented approach extends and improves different aspects of the previous methodology. By deriving velocity constraints from the feet that are in contact with the ground, simple measurement equations are obtained which reduce the size of the state and which are more suitable for slippage detection. Further, a robot-centric formulation of the state space is chosen in order to appropriately partition the filter states and avoid problems with unobservable states.

A thorough nonlinear observability analysis is provided for the presented filter. A novel method for handling rotational states is presented which significantly simplifies the analytical evaluation of the unobservable subspace and corresponding rank deficiency. Based on the nonlinear observability analysis of Hermann and Krener [57] we present a method for handling states which are elements of the special orthogonal group $SO(3)$ by exploiting the local homeomorphism to 3D real vector space. With this we show, that up to some singular robot motions, all states of the robotic platform are observable except for the yaw angle around the gravity axis and the global position (which are not essential for the local control of the robot). This also holds for the case where only one leg is in contact with the ground and thus the state estimator can be applied for dynamic locomotion as well.

The presented filter is implemented and evaluated on our quadruped robot StarlETH [63]. We show results from experiments where the robot is trotting over uneven and labile terrain with occurring foot slippage. For all experiments the control of the robot fully relies on the estimates from the filter. No previous information on the shape of the terrain is required and the external motion capture system is only used for groundtruth comparison.

The paper is structured as follows. In Section 2 we start with some brief prerequisites. Subsequently, Section 3 discusses the specific filter setup and the outliers detection. In Section 4 the observability analysis is performed and in Section 5 the experimental setup and obtained results are presented.

2 Prerequisites

For better readability we give a short overview on the employed notations and conventions. The coordinates, expressed in a frame A, of a vector from a point P to a point Q are denoted by ${}_A r_{PQ}$. If B is a second coordinate frame, then $\boldsymbol{C}_{BA}$ maps the coordinates expressed in A to the corresponding coordinates in B. The rotation between both frames is generally parametrized by the unit quaternion $\boldsymbol{q}_{BA}$. Throughout the paper, we add a subscript k to a quantity v, if we want to talk about its value at a time t_k, i.e., $v_k = v(t_k)$. Two coordinate frames are of interest: the world fixed

coordinate frame W and the main body frame B.

In a filter setup, mathematical operations are employed which are not defined for 3D rotations (especially addition and differentiation). In order to handle this issue we exploit the homeomorphism between the 3D manifold $SO(3)$ and 3D vector spaces. For a more thorough discussion on the topic please refer to the work of Hertzberg et al. [59]. In short we use the exponential mapping, $\boldsymbol{q} = \exp(\boldsymbol{\theta})$, between a 3D rotation vector, $\boldsymbol{\theta} \in \mathbb{R}^3$, and the corresponding quaternion $\boldsymbol{q} \in SO(3)$. This mapping is surjective and thus an inverse exists, $\boldsymbol{\theta} = \log(\boldsymbol{q})$, which is called the logarithm. These maps are used for introducing the boxplus and boxminus operators:

$$\begin{aligned} \boxplus : & SO(3) \times \mathbb{R}^3 \to SO(3), \\ & \boldsymbol{q}, \boldsymbol{\theta} \mapsto \exp(\boldsymbol{\theta}) \otimes \boldsymbol{q}, \end{aligned} \tag{8.1}$$

$$\begin{aligned} \boxminus : & SO(3) \times SO(3) \to \mathbb{R}^3, \\ & \boldsymbol{q}_1, \boldsymbol{q}_2 \mapsto log(\boldsymbol{q}_1 \otimes \boldsymbol{q}_2^{-1}), \end{aligned} \tag{8.2}$$

where the boxminus operator expresses the difference between two quaternions by returning the error rotation vector between both, and where the boxplus operator applies a small rotation, expressed by a rotation vector, onto a unit quaternion.

Based on the above definitions we introduce special differentials on unit quaternions. Given a function $\boldsymbol{q} : \boldsymbol{x} \mapsto \boldsymbol{q}(\boldsymbol{x})$ which maps from some real vector space $\mathbb{R}^N$ to the set of unit quaternions, we define the differential

$$\left(\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}}\right)_i := \lim_{\epsilon \to 0} \frac{\boldsymbol{q}(\boldsymbol{x} + \epsilon \boldsymbol{e}_i) \boxminus \boldsymbol{q}(\boldsymbol{x})}{\epsilon}, \quad i = 1, \ldots, N, \tag{8.3}$$

and if $\boldsymbol{f} : \boldsymbol{q} \mapsto \boldsymbol{f}(\boldsymbol{q})$ is a function which maps from the set of unit quaternions to some real vector space we define

$$\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}\right)_i := \lim_{\epsilon \to 0} \frac{\boldsymbol{f}(\boldsymbol{q} \boxplus \epsilon \boldsymbol{e}_i) - \boldsymbol{f}(\boldsymbol{q})}{\epsilon}, \quad i = 1, \ldots, 3. \tag{8.4}$$

Let $\boldsymbol{C}(\cdot)$ be the mapping between unit quaternions and corresponding rotation matrices, then following identities hold:

$$\partial / \partial \boldsymbol{q} \left(\boldsymbol{C}(\boldsymbol{q}) \boldsymbol{v} \right) = - \left(\boldsymbol{C}(\boldsymbol{q}) \boldsymbol{v} \right)^{\times}, \tag{8.5}$$

$$\partial / \partial \boldsymbol{q} \left(\boldsymbol{q}^{-1} \right) = -\boldsymbol{C}^T(\boldsymbol{q}), \tag{8.6}$$

$$\partial / \partial \boldsymbol{q}_1 \left(\boldsymbol{q}_1 \otimes \boldsymbol{q}_2 \right) = \boldsymbol{I}, \tag{8.7}$$

$$\partial / \partial \boldsymbol{q}_2 \left(\boldsymbol{q}_1 \otimes \boldsymbol{q}_2 \right) = \boldsymbol{C}(\boldsymbol{q}_1), \tag{8.8}$$

$$\partial / \partial \boldsymbol{q} \left(\log(\boldsymbol{q}) \right) = \boldsymbol{\Gamma}_1^{-1}(\log(\boldsymbol{q})), \tag{8.9}$$

$$\partial / \partial t \left(\boldsymbol{q}_{BA}(t) \right) = {}_B\boldsymbol{\omega}_{BA}(t), \tag{8.10}$$

where the subscript $^{\times}$ is used to denote the skew-symmetric matrix of a vector and where $\boldsymbol{\omega}_{BA}$ is the rotational rate vector of frame B with respect to frame A. We also made use of the auxiliary quantity $\boldsymbol{\Gamma}_n(\boldsymbol{\theta}) := \sum_{i=0}^{\infty} \frac{\boldsymbol{\theta}^{\times^i}}{(i+n)!}$. It draws on the series

expansion of the matrix exponential and, consequently, $\boldsymbol{\Gamma}_0(\boldsymbol{\theta})$ represents the rotation matrix corresponding to the rotation vector $\boldsymbol{\theta}$. There exists a closed form expression for $\boldsymbol{\Gamma}_n$ that can be efficiently numerically evaluated (similar to Rodrigues' rotation formula). The above special differentials strongly simplify the handling of analytical Jacobians, especially in the context of nonlinear observability analysis including rotational quantities. It can be proven that the chain rule is valid. Please note that the formulation of the identities can vary slightly depending on the employed conventions.

3 Filter Setup

3.1 Filter States and Measurement Models

The overall structure of a filter strongly depends on the choice of the underlying filter states. In our case we chose a set of robot-centric states in order to describe the motion of the robot's main body. The state includes the position of the world frame with respect to the body frame, ${}_B\boldsymbol{r}_{BW}$, the negative velocity of the main body expressed in the frame B, $-{}_B\boldsymbol{v}_B$, the attitude of the main body parametrized by $\boldsymbol{q}_{WB}$, as well as the bias terms of the accelerometer and gyroscope, ${}_B\boldsymbol{b}_f$ and ${}_B\boldsymbol{b}_\omega$. In short, the state $\boldsymbol{x}$ will be defined as

$$\begin{aligned}\boldsymbol{x} &:= \left(\boldsymbol{r}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{c}, \boldsymbol{d}\right) && (8.11)\\ &:= \left({}_B\boldsymbol{r}_{BW}, -{}_B\boldsymbol{v}_B, \boldsymbol{q}_{WB}, {}_B\boldsymbol{b}_f, {}_B\boldsymbol{b}_\omega\right). && (8.12)\end{aligned}$$

Building on this, process and measurement equations need to be formulated which properly capture the behavior and uncertainties of the underlying system. The choice of the models is a trade-off between simplicity and accuracy, whereby all stochastic quantities will be modeled as continuous white Gaussian noise or as discrete Gaussian noise processes. This is in accord with the prerequisites of most filtering methods and deviation from the real system can be handled to a certain extent by increasing the corresponding covariance matrices.

The proper acceleration measurement $\tilde{\boldsymbol{f}}$ and the rotational rate measurement $\tilde{\boldsymbol{\omega}}$ of the IMU are assumed to be affected by additive white Gaussian noise, $\boldsymbol{n}_f$ and $\boldsymbol{n}_\omega$, as well as by the additive biases $\boldsymbol{c}$ and $\boldsymbol{d}$:

$$\begin{aligned}\tilde{\boldsymbol{f}} &= \boldsymbol{f} + \boldsymbol{c} + \boldsymbol{n}_f, && (8.13)\\ \tilde{\boldsymbol{\omega}} &= \boldsymbol{\omega} + \boldsymbol{d} + \boldsymbol{n}_\omega. && (8.14)\end{aligned}$$

Both quantities do not directly depend on the states of the filter but rather measure the corresponding rates. Considering

$$\begin{aligned}\boldsymbol{f} &= \boldsymbol{C}(\boldsymbol{q}_{BW})\left({}_W\dot{\boldsymbol{v}}_B - \boldsymbol{g}\right), && (8.15)\\ \boldsymbol{\omega} &= \dot{\boldsymbol{q}}_{BW}, && (8.16)\end{aligned}$$

where $\boldsymbol{g}$ is the gravity vector in W, the IMU measurements will later be directly included into the prediction step of the filter. For simplicity, we assume that all inertial measurements are obtained with respect to the body frame B.

Encoders in each of the robot's joints provide access to the corresponding angular measurements $\tilde{\boldsymbol{\alpha}}$ and their derivatives $\dot{\tilde{\boldsymbol{\alpha}}}$. Considering the forward kinematics ${}_B\boldsymbol{r}_{BF_i}(\tilde{\boldsymbol{\alpha}}) = \boldsymbol{s}_i(\tilde{\boldsymbol{\alpha}})$, we can compute the absolute location of the i^{th} foot F_i:

$$
\begin{aligned}
{}_W\boldsymbol{r}_{WF_i} &= {}_W\boldsymbol{r}_{WB} + \boldsymbol{C}_{WB}\,{}_B\boldsymbol{r}_{BF_i}(\tilde{\boldsymbol{\alpha}}) && (8.17)\\
&= \boldsymbol{C}(\boldsymbol{q})(\boldsymbol{s}_i(\tilde{\boldsymbol{\alpha}}) - \boldsymbol{r}). && (8.18)
\end{aligned}
$$

If foot i is in contact with the ground and assuming that it remains stationary with respect to the world frame W, the differentiation of the above kinematic identity yields

$$
0 = -\boldsymbol{v} + \boldsymbol{\omega}^{\times}\boldsymbol{s}_i(\tilde{\boldsymbol{\alpha}}) + \boldsymbol{J}_i(\tilde{\boldsymbol{\alpha}})\dot{\tilde{\boldsymbol{\alpha}}} + \boldsymbol{n}_s, \tag{8.19}
$$

where $\boldsymbol{J}_i(\tilde{\boldsymbol{\alpha}}) = \frac{\partial}{\partial\tilde{\boldsymbol{\alpha}}}\boldsymbol{s}_i(\tilde{\boldsymbol{\alpha}})$ is the Jacobian of the forward kinematics. The discrete Gaussian noise term $\boldsymbol{n}_s \sim \mathcal{N}(0, \boldsymbol{R}_s)$ incorporates different sources of noise, including errors from the encoder measurements as well as imprecise kinematic modeling. This is mainly done because the noise on the encoder measurements causes only a minor part of the full measurement noise of (8.19), where modeling errors and foot contact effects are more important. In order to avoid the complex modeling of such effects, the covariance matrix $\boldsymbol{R}_s$ incorporates all stochastic errors together and represents one of the main tuning parameter of the filter.

As mentioned earlier, the IMU measurements are linked to the rates of the filter states and are thus included into the continuous time differential equations of the prediction model. Using equation (8.15) and (8.16) and carefully evaluating the total derivatives we can write:

$$
\begin{aligned}
\dot{\boldsymbol{r}} &= -(\tilde{\boldsymbol{\omega}} - \boldsymbol{d} - \boldsymbol{n}_\omega)^{\times}\boldsymbol{r} + \boldsymbol{v}, && (8.20)\\
\dot{\boldsymbol{v}} &= -(\tilde{\boldsymbol{\omega}} - \boldsymbol{d} - \boldsymbol{n}_\omega)^{\times}\boldsymbol{v} - \tilde{\boldsymbol{f}} + \boldsymbol{c} + \boldsymbol{n}_f - \boldsymbol{C}^T(\boldsymbol{q})\boldsymbol{g}, && (8.21)\\
\dot{\boldsymbol{q}} &= \boldsymbol{C}(\boldsymbol{q})(\tilde{\boldsymbol{\omega}} - \boldsymbol{d} - \boldsymbol{n}_\omega), && (8.22)\\
\dot{\boldsymbol{c}} &= \boldsymbol{n}_c, && (8.23)\\
\dot{\boldsymbol{d}} &= \boldsymbol{n}_d. && (8.24)
\end{aligned}
$$

The additional continuous white Gaussian noise processes $\boldsymbol{n}_c$ and $\boldsymbol{n}_d$ model a certain drift affecting the bias terms. For all white Gaussian noise processes, the corresponding covariance parameters, $\boldsymbol{R}_f$, $\boldsymbol{R}_\omega$, $\boldsymbol{R}_c$ and $\boldsymbol{R}_d$ describe the magnitude of the noise. The covariance parameters can be identified by considering the Allan plots of the IMU measurements [32].

3.2 Unscented Kalman Filter

The different measurements are fused within an unscented Kalman filter framework. While the resulting computational costs are slightly higher than for a corresponding extended Kalman filter, the UKF is in general more robust against nonlinearities. However, for the case at hand, our choice was mainly motivated by the simplicity of handling correlated noise between prediction and correction step. The correlation can best be seen by considering the discretized filter equations.

Discretization of the stochastic differential equations (SDE) (8.20)-(8.24) is a difficult problem and is, in general, not analytically solvable without approximation. The most common approach is to linearize the equations and to integrate the linear SDE. Here, we discretize the deterministic and stochastic part of the SDE separately. This allows the analytical solution of the corresponding system of deterministic differential equations and thus keeps our rotational state in the 3D manifold $SO(3)$. Using the abbreviation $\Delta t_k = t_k - t_{k-1}$ and applying the method of variation of parameters we obtain:

$$\begin{aligned}\boldsymbol{r}_k = \boldsymbol{\Gamma}_{0,k}^T \Big(& \boldsymbol{r}_{k-1} + \Delta t_k \boldsymbol{v}_{k-1} - \frac{\Delta t_k^2}{2} \Big(2\boldsymbol{\Gamma}_{2,k} (\tilde{\boldsymbol{f}}_k \\ & - \boldsymbol{c}_{k-1} - \boldsymbol{n}_{f,k}) + \boldsymbol{C}(\boldsymbol{q}_{k-1}) \boldsymbol{g} \Big) \Big) + \boldsymbol{n}_{r,k}, \end{aligned} \tag{8.25}$$

$$\begin{aligned}\boldsymbol{v}_k = \boldsymbol{\Gamma}_{0,k}^T \Big(& \boldsymbol{v}_{k-1} - \Delta t_k \Big(\boldsymbol{\Gamma}_{1,k} (\tilde{\boldsymbol{f}}_k - \boldsymbol{c}_{k-1} \\ & - \boldsymbol{n}_{f,k}) + \boldsymbol{C}(\boldsymbol{q}_{k-1}) \boldsymbol{g} \Big) \Big), \end{aligned} \tag{8.26}$$

$$\boldsymbol{q}_k = \boldsymbol{q}_{k-1} \otimes \exp\Big(\Delta t_k (\tilde{\boldsymbol{\omega}}_k - \boldsymbol{d}_{k-1} - \boldsymbol{n}_{\omega,k}) \Big), \tag{8.27}$$

$$\boldsymbol{c}_k = \boldsymbol{c}_{k-1} + \Delta t_k \boldsymbol{n}_{c,k}, \tag{8.28}$$

$$\boldsymbol{d}_k = \boldsymbol{d}_{k-1} + \Delta t_k \boldsymbol{n}_{d,k}, \tag{8.29}$$

with

$$\boldsymbol{\Gamma}_{n,k} = \boldsymbol{\Gamma}_n \Big(\Delta t_k (\tilde{\boldsymbol{\omega}}_k - \boldsymbol{d}_{k-1} - \boldsymbol{n}_{\omega,k}) \Big). \tag{8.30}$$

The various discretized noise quantities are distributed with $\mathcal{N}(0, \boldsymbol{R}/\Delta t_k)$ where $\boldsymbol{R}$ is the corresponding continuous covariance parameter. The new discrete Gaussian noise term $\boldsymbol{n}_{r,k}$ is used to model errors that occurred during discretization.

While equations (8.25)-(8.29) are used for the prediction of the filter, the update step is based on the kinematic identity (8.19). This is applied to every leg i that is in contact:

$$\begin{aligned} 0 = & -\boldsymbol{v}_k + (\tilde{\boldsymbol{\omega}}_k - \boldsymbol{d}_{k-1} - \boldsymbol{n}_{\omega,k})^\times \boldsymbol{s}_i(\tilde{\boldsymbol{\alpha}}_k) \\ & + \boldsymbol{J}_i(\tilde{\boldsymbol{\alpha}}_k) \dot{\tilde{\boldsymbol{\alpha}}}_k + \boldsymbol{n}_{s,k}. \end{aligned} \tag{8.31}$$

The recurrence of the gyroscope measurement noise $\boldsymbol{n}_{\omega,k}$ in the update equation correlates the noise between prediction and update step. In an UKF setup this can be handled very easily. The basic outline of the filter looks as follows. Given the *a-posteriori* estimate $\boldsymbol{x}_{k-1}$ and its covariance matrix $\boldsymbol{P}_{k-1}$ at time t_{k-1}, sigma points are sampled in such a manner that they represent the joint distribution of the state estimate and all noise quantities. This results in a set of sigma points of the following form:

$$\mathcal{X}_{k-1}^i = \Big(\boldsymbol{x}_{k-1}^i, \boldsymbol{n}_{r,k}^i, \boldsymbol{n}_{f,k}^i, \boldsymbol{n}_{\omega,k}^i, \boldsymbol{n}_{c,k}^i, \boldsymbol{n}_{d,k}^i, \boldsymbol{n}_{s,k}^i \Big). \tag{8.32}$$

Whereby using the same sampled rotational rate noise during prediction and update automatically handles the stochastic correlation between both steps. For a more detailed discussion on the employed UKF please refer to [74]. Also, please note that throughout the filter the boxplus (8.1) and boxminus (8.2) operators have to be employed where appropriate.

3.3 Outliers Detection

Kalman filters have the drawback that they can be very sensitive to outliers. While outliers are often caused by non-modeled effects or other anomalies, their appearance is in most cases only difficultly predictable and the corresponding observations draw generally from a significantly different probability distribution. The sensitivity is caused by the light-tailed underlying Gaussian distribution which leads to the minimization of squared error terms. In order to handle outliers caused by foot slippage we propose to employ a simple thresholding based on the Mahalanobis distance of the innovation. This employs the predicted covariance of the innovation and classifies a measurement as an outlier if the Mahalanobis distance exceeds a certain threshold. This has the drawback that the threshold needs to be hand-tuned, however, if it is appropriately chosen this leads to near-optimal filtering [133].

Let $\boldsymbol{y}_{i,k}$ be the innovation induced by the kinematic constraints of the i^{th} leg at timestep k (8.31) and $\boldsymbol{S}_{i,k}$ the corresponding predicted covariance matrix. We classify the observation as an outlier if the Mahalanobis distance is larger than a certain threshold parameter p, i.e., if $\boldsymbol{y}_{i,k}^T \boldsymbol{S}_{i,k}^{-1} \boldsymbol{y}_{i,k} > p$ is met. Under the assumption of Gaussian distribution the left hand side of the inequality will be χ^2 distributed with 3 degrees of freedom. In our case the threshold $p = 16.27$ was chosen in order to obtain a rejection rate of 0.1% for inliers. If the above threshold is exceeded, the kinematic constraints are ignored and not taken into account during the update step (like for all legs that are not currently in contact with the ground). An analogous approach was employed by Mirzaei et al. [96] for rejecting visual feature measurements within a Kalman filter based IMU-camera calibration.

4 Nonlinear Observability Analysis

Similarly to Hermann and Krener [57], we employ the notion of locally weakly observability which qualifies whether each point of a system can be instantaneously distinguished from its neighbors. As a slight technical difference we consider our system to have no external control input and interpret the rotational rate as well as the proper acceleration as system parameters. The subsequent nonlinear observability analysis should reflect the observability characteristics of the system in dependence of those parameters.

Lets consider the following state-space representation of a smooth nonlinear system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}), \tag{8.33}$$

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}), \tag{8.34}$$

with process function $\boldsymbol{f}$ and measurement function $\boldsymbol{h}$. For a given state $\boldsymbol{x}$ and input parameters $\boldsymbol{u}$ we can now evaluate the observability matrix

$$\boldsymbol{\mathcal{O}}(\boldsymbol{x},\boldsymbol{u}) = \begin{bmatrix} \nabla \mathcal{L}_{\boldsymbol{f}}^{0} \boldsymbol{h}(\boldsymbol{x}) \\ \nabla \mathcal{L}_{\boldsymbol{f}}^{1} \boldsymbol{h}(\boldsymbol{x}) \\ \vdots \end{bmatrix}, \tag{8.35}$$

based on the gradient operator ∇ and Lie derivatives [57]. Informally it describes the effect of infinitesimal state perturbations $\boldsymbol{\delta x}$ on the instantaneous measurement $\boldsymbol{z}$ and its derivatives:

$$\begin{bmatrix} \boldsymbol{\delta z} \\ \boldsymbol{\delta \dot{z}} \\ \vdots \end{bmatrix} = \boldsymbol{\mathcal{O}}(\boldsymbol{x},\boldsymbol{u}) \boldsymbol{\delta x}. \tag{8.36}$$

Perturbations $\boldsymbol{\delta x}$ which do not cause any change in the corresponding measurements are intrinsically not observable. Consequently, the nullspace of the observability matrix $\boldsymbol{\mathcal{O}}(\boldsymbol{x},\boldsymbol{u})$ is equivalent to the unobservable subspace of the system at a state $\boldsymbol{x}$ and for a given input parameter $\boldsymbol{u}$.

The novelty in the presented nonlinear observability analysis is the seamless integration of rotational states into the observability analysis by means of the special derivatives introduced in equations (8.3) and (8.4). Using the identities (8.5)-(8.10) and applying the chain rule, the Lie derivatives can be easily evaluated, whereby the entries in the observability matrix corresponding to 3D rotational quantities will exhibit the proper number of dimension (which should be 3) and accurately reflect the observability characteristics of the system. This is best explained at hand of a concrete example: for the filter presented in this paper the sequence of Lie derivatives and corresponding gradients together with the observability matrix will be evaluated for the case of a single foot contact with the ground. For the sake of readability the indexes are omitted where possible and the noise terms are left out (they do not influence the observability analysis). In short we will also use $\tilde{\boldsymbol{s}} = \boldsymbol{s}(\tilde{\boldsymbol{\alpha}})$, $\hat{\boldsymbol{\omega}} = \tilde{\boldsymbol{\omega}} - \boldsymbol{d}$, $\hat{\boldsymbol{f}} = \tilde{\boldsymbol{f}} - \boldsymbol{c}$ and $\boldsymbol{C} = \boldsymbol{C}(\boldsymbol{q})$. The process function (equations (8.20)-(8.24)) can be written as:

$$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) = \begin{pmatrix} -\hat{\boldsymbol{\omega}}^{\times} \boldsymbol{r} + \boldsymbol{v} \\ -\hat{\boldsymbol{\omega}}^{\times} \boldsymbol{v} - \hat{\boldsymbol{f}} - \boldsymbol{C}^{T} \boldsymbol{g} \\ \boldsymbol{C} \hat{\boldsymbol{\omega}} \\ 0 \\ 0 \end{pmatrix}. \tag{8.37}$$

The sequence of Lie derivatives and corresponding gradients can be evaluated to:

$$\mathcal{L}_{\boldsymbol{f}}^{0}\boldsymbol{h}(\boldsymbol{x}) = -\boldsymbol{v} + (\tilde{\boldsymbol{\omega}} - \boldsymbol{d})^{\times}\tilde{\boldsymbol{s}} + \boldsymbol{J}(\tilde{\boldsymbol{\alpha}})\dot{\tilde{\boldsymbol{\alpha}}}, \tag{8.38}$$

$$\nabla\mathcal{L}_{\boldsymbol{f}}^{0}\boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix} 0 & -\boldsymbol{I} & 0 & 0 & \tilde{\boldsymbol{s}}^{\times} \end{bmatrix}, \tag{8.39}$$

$$\mathcal{L}_{\boldsymbol{f}}^{1}\boldsymbol{h}(\boldsymbol{x}) = \hat{\boldsymbol{\omega}}^{\times}\boldsymbol{v} + \hat{\boldsymbol{f}} + \boldsymbol{C}^{T}\boldsymbol{g}, \tag{8.40}$$

$$\nabla\mathcal{L}_{\boldsymbol{f}}^{1}\boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix} 0 & \hat{\boldsymbol{\omega}}^{\times} & \boldsymbol{C}^{T}\boldsymbol{g}^{\times} & -\boldsymbol{I} & \boldsymbol{v}^{\times} \end{bmatrix}, \tag{8.41}$$

$$\vdots$$

$$\mathcal{L}_{\boldsymbol{f}}^{n}\boldsymbol{h}(\boldsymbol{x}) = -\hat{\boldsymbol{\omega}}^{\times^{n}}\boldsymbol{v} - \hat{\boldsymbol{\omega}}^{\times^{n-1}}\hat{\boldsymbol{f}} - n\hat{\boldsymbol{\omega}}^{\times^{n-1}}\boldsymbol{C}^{T}\boldsymbol{g}, \tag{8.42}$$

$$\nabla\mathcal{L}_{\boldsymbol{f}}^{n}\boldsymbol{h}(\boldsymbol{x}) = \Big[0 \quad -\hat{\boldsymbol{\omega}}^{\times^{n}} \quad -n\hat{\boldsymbol{\omega}}^{\times^{n-1}}\boldsymbol{C}^{T}\boldsymbol{g}^{\times} \quad \hat{\boldsymbol{\omega}}^{\times^{n-1}} \quad \partial\mathcal{L}_{\boldsymbol{f}}^{n}\boldsymbol{h}(\boldsymbol{x})/\partial\boldsymbol{d}\Big]. \tag{8.43}$$

With this, the Observability matrix (8.35) can be constructed and simplified in order to obtain the following term:

$$\boldsymbol{\mathcal{O}}(\boldsymbol{x},\boldsymbol{u}) = \begin{bmatrix} 0 & -\boldsymbol{I} & 0 & 0 & \tilde{\boldsymbol{s}}^{\times} \\ 0 & 0 & \boldsymbol{C}^{T}\boldsymbol{g}^{\times} & -\boldsymbol{I} & \boldsymbol{v}^{\times} + \hat{\boldsymbol{\omega}}^{\times}\tilde{\boldsymbol{s}}^{\times} \\ 0 & 0 & -\hat{\boldsymbol{\omega}}^{\times}\boldsymbol{C}^{T}\boldsymbol{g}^{\times} & 0 & (\hat{\boldsymbol{\omega}}^{\times}\boldsymbol{v} + \hat{\boldsymbol{f}} + 2\boldsymbol{C}^{T}\boldsymbol{g})^{\times} \\ 0 & 0 & 0 & 0 & (\hat{\boldsymbol{\omega}}^{\times}\boldsymbol{v} + \hat{\boldsymbol{f}} + \boldsymbol{C}^{T}\boldsymbol{g})^{\times}\hat{\boldsymbol{\omega}}^{\times} \\ 0 & 0 & 0 & 0 & (\hat{\boldsymbol{\omega}}^{\times}\boldsymbol{v} + \hat{\boldsymbol{f}} + \boldsymbol{C}^{T}\boldsymbol{g})^{\times}\hat{\boldsymbol{\omega}}^{\times 2} \\ 0 & 0 & 0 & 0 & (\hat{\boldsymbol{\omega}}^{\times}\boldsymbol{g})^{\times} \\ 0 & 0 & 0 & 0 & (\hat{\boldsymbol{\omega}}^{\times 2}\boldsymbol{g})^{\times} \end{bmatrix}$$

In this example, the input parameter $\boldsymbol{u}$ is given by the rotational rate $\hat{\boldsymbol{\omega}}$ and the proper acceleration $\hat{\boldsymbol{f}}$ which describe the motion of the robot main body. Our goal is to obtain the observability characteristic in dependence of those parameters, rather than asking the question whether there exists some input parameter which make our system observable.

As mentioned above, the nullspace of the observability matrix corresponds to the directions of disturbances which can not be observed at the output of the system. Up to a few singular cases, the rank of the nullspace is 4 and is spanned by the following matrix:

$$\boldsymbol{\mathcal{U}}(\boldsymbol{x},\boldsymbol{u}) = \begin{bmatrix} \boldsymbol{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & \boldsymbol{g}^{T} & 0 & 0 \end{bmatrix}^{T}, \tag{8.44}$$

where the first row describes unobservable disturbances on the robot position and where the second row represents rotation around the gravity axis (yaw angle). The emergence of those unobservable modes could have been predicted as we do not use any global positioning system. However, there are singular cases where more directions become unobservable. Those can also be evaluated analytically based on the

$\hat{\boldsymbol{\omega}} = 0$	$\hat{\boldsymbol{\omega}} \perp \boldsymbol{C}^T\boldsymbol{g}$	$\hat{\boldsymbol{\omega}} \parallel \boldsymbol{C}^T\boldsymbol{g}$	Rank deficiency
x	x	x	5 $(\hat{\boldsymbol{f}} = -2\boldsymbol{C}^T\boldsymbol{g})$ 3 $(\hat{\boldsymbol{f}} \neq -2\boldsymbol{C}^T\boldsymbol{g})$
	x		1
		x	1
			0

Table 8.1: Rank deficiency in dependence of input parameters.

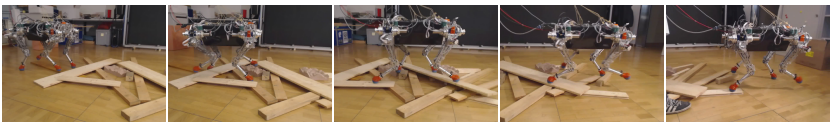

Figure 8.1: Trotting sequence over uneven and slippery terrain. The robot requires about 15 s for traversing the 3 m long area covered with loose wooden planks.

observability matrix. In the scope of this paper the singular cases are listed in table 8.1 together with a brief discussion (if a cross is set the above equality is fulfilled). As can be observed, the rank loss depends on the relation between gravity vector and rotational rate vector. If there is no rotational motion in the system, the filter can not distinguish between inclination angles (pitch and roll) and a bias on the proper acceleration measurement. Furthermore it will not be able to estimate the gyroscope bias around the gravity and we thus get a total rank deficiency of 3 for this case.

In a less intuitive way the system loses two further ranks if it does not exhibit any rotational motion and, at the same time, accelerates with $-\boldsymbol{g}$ in the world frame ($\hat{\boldsymbol{f}} = -2\boldsymbol{C}^T\boldsymbol{g}$). This represents a rather unrealistic situation our robot might find itself in. In the general case the system will rarely be perfectly at a singular point and thus the corresponding filter should be able to observe all state except for the globally unobservable position and yaw angle. Also, please remember that the above table describes the case where only a single foot is in contact with the ground and that the rank deficiency tends to be smaller if more contacts are available.

5 Results and Discussion

The presented filter was implemented and evaluated on our quadruped platform StarlETH [63]. For the experiments the output of the state estimation was used to stabilize and control the robot. We illustrate the filter performance at hand of an experiment where the robot trots over uneven and highly slippery terrain. Figure 8.1 shows a sequence of images depicting the trajectory of the robot. It covers a distance of approximately 3 m in roughly 15 s.

In Figure 8.2 a detailed sequence of snapshots shows a slip situation towards the end of the experiment (around the last image of Figure 8.1). For this sequence we

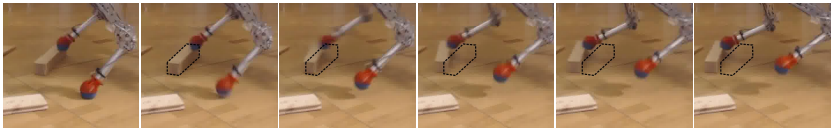

Figure 8.2: A sequence of snapshots illustrating the substantial slip that is occurring during the experiment. If looking at the plank beneath the left foot one can observe that it is moved by approximately 10 cm. Time between snapshots: 32 ms.

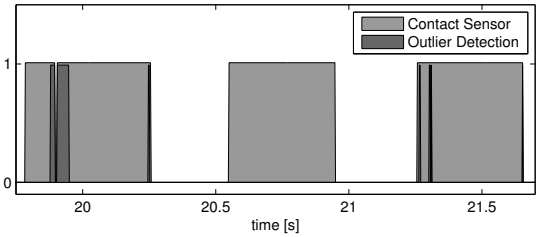

Figure 8.3: Binary outputs from contact sensor and outlier detection of the left hind leg. Light gray: flag of contact sensor (1 = contact). Dark grey: outlier detection (only detect outliers if the contact sensor flag is true). Three stance phases are displayed. In the first stance phase slippage is detected which corresponding to the slip event illustrated in Figure 8.2.

plotted the results of the outlier detection algorithm of Section 3.3 in Figure 8.3. The three distinct blocks in the figure correspond to the stance phases of three subsequent steps of the left hind leg. While the light gray surface represents the contact detected by the contact sensor, the dark gray surfaces represent the detection of outliers. The first block corresponds to the slip situation of Figure 8.2, where the dark gray phase towards the beginning of the stance phase represents detected slip (the contact is also lost for a very short instant). There are a few unexpected outlier detections throughout the dataset. They often occur at the beginning or towards the end of stance phase where the foot is not well in contact with the ground and where oscillations can occur due to the compliance of the foot. In contrast to our previous work [10], where an estimate of the foothold is initialized at each new step, the present filter is much less susceptible to fast switching foot contacts.

Figure 8.4 and Figure 8.5 show the resulting estimates for the attitude and the velocity of the robot main body. From the point of view of the local controller those quantities are of high importance in order to enable the stabilization of the main body. As pointed out in Section 4 the angle around the gravity axis (yaw) is not observable and consequently the filter estimate will drift away. However, for the remaining two

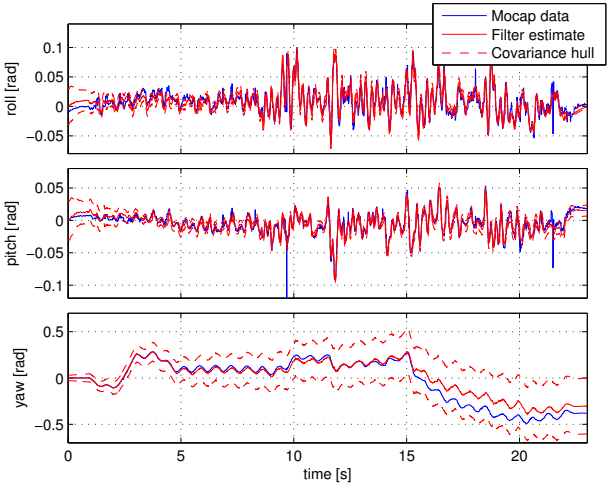

Figure 8.4: Roll, pitch, and yaw angles of the main body for the sequence depicted in Figure 8.1. Red: estimated values. Red dashes: 3σ covariance bounds. Blue: motion capture data. The RMS values for the roll, pitch, and yaw estimates are: 0.0086 rad, 0.0056 rad, 0.0693 rad.

degrees of freedom (pitch and roll) very precise results are obtained with RMS values below 0.01 rad if compared to the motion capture data. The plotted 3σ covariance bounds of the attitude estimates roughly captures the uncertainty of the system and the motion capture attitude remains between the bounds for most of the time (there are some outliers in the motion capture data).

The velocity estimates are more difficult to evaluate due to noisy numerical differentials of the motion capture system. Still, one can observe a nice overlay between both trajectories. Here, all three quantities are observable and after a very quick initial convergence the covariance estimates remain more or less constant. The obtained RMS values are around 0.05 m/s, whereas a large amount is caused by the noisy motion capture estimates. If compared to the filter presented in [10] the RMS errors for the velocity estimates as well as for the roll and pitch angles are roughly halved for this experiment (for pitch there is even a factor 10). This comes at costs of accuracy on the position and yaw angle. However, as mentioned earlier those quantities are of secondary interest and their estimation could be improved by integrating more suitable sensor modalities like vision or LIDAR.

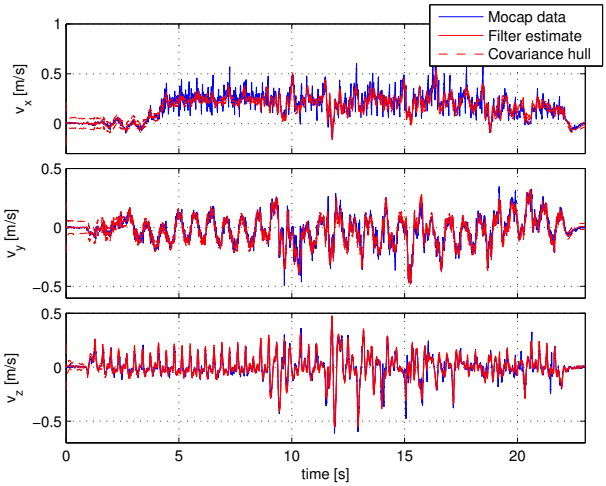

Figure 8.5: Velocity estimates of the main body for the sequence depicted in Figure 8.1. Red: estimated velocity values. Red dashes: 3σ covariance bounds. Blue: motion capture data. The RMS values for the three velocity estimates are: 0.0546 m/s, 0.0406 m/s, 0.0348 m/s.

6 Conclusion and Future Work

In this paper we presented a novel state estimation approach for legged robots based on kinematic velocity measurements at the ground contacts. The obtained information is fused with measurements from an on-board IMU by means of an unscented Kalman filter. The provided nonlinear observability analysis shows that, for general robot motions, all states are observable except for the global position and the yaw angle. This results in a filter which accurately estimates the inclination angles (roll and pitch) as well as the velocities of the robot. It also avoids unnecessary assumptions on the shape of the floor or on the employed gait pattern and is robust to a certain amount of foot slippage. Implemented on our legged robot StarlETH, it enables dynamic locomotion over uneven and labile terrain.

While the position and the yaw angle of the robot are quantities which are less critical for a local stabilization of force controlled legged robots, they are important for global navigation. Future work will thus include evaluating different methods for integrating further sensor modalities which are more suited for navigation and terrain perception.

Paper V

Fusion of Optical Flow and Inertial Measurements for Robust Egomotion Estimation

Michael Bloesch, Sammy Omari, Peter Fankhauser, Hannes Sommer, Christian Gehring, Jemin Hwangbo, Mark A. Hoepflinger, Marco Hutter, Roland Siegwart

Abstract

In this paper we present a method for fusing optical flow and inertial measurements. To this end, we derive a novel visual error term which is better suited than the standard continuous epipolar constraint for extracting the information contained in the optical flow measurements. By means of an unscented Kalman filter (UKF), this information is then tightly coupled with inertial measurements in order to estimate the egomotion of the sensor setup. The individual visual landmark positions are not part of the filter state anymore. Thus, the dimensionality of the state space is significantly reduced, allowing for a fast online implementation. A nonlinear observability analysis is provided and supports the proposed method from a theoretical side. The filter is evaluated on real data together with ground truth from a motion capture system.

Published in:
IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014
DOI: 10.1109/IROS.2014.6942991

1 Introduction

The use of cameras as light-weight egomotion sensors has been studied very broadly in the past few decades. The main advantage of a camera is that rich information can be obtained at relatively low power consumption. However, this information richness also poses the main difficulty, as the vast amount of information needs to be handled properly before the egomotion can be inferred.

Within the computer vision community, Davison [28] presented one of the first algorithms that is able to accurately track the 3D pose of a monocular camera. His idea was to design an Extended Kalman Filter (EKF) which simultaneously tracks the pose of the camera as well as the 3D position of points of interest, whereby the reprojection errors of the perceived features serve as innovation term. In the following, different authors presented adaptations in order to tackle different weaknesses of this approach, such as feature initialization [98] and limited map size [26].

Compared to the above mentioned *non-delayed* approaches, *delayed* methods also take past robot poses and measurements into account. The delayed approaches have become popular with the work of Klein and Murray [81]: Based on a subset of camera frames (keyframes) a bundle adjustment algorithm [135] optimizes a map, while the actual pose of the camera is tracked by minimizing the reprojection error between map and camera. Strasdat et al. [130] argued that in terms of accuracy and computational costs it would be more beneficial to increase the number of tracked features rather than the number of frames they are tracked in. In the following, the limits of vision-only state estimation and mapping where pushed even further by various other elaborate delayed frameworks [76, 95, 131].

In parallel to the "vision-only" based approaches, other researchers started including inertial measurements into their estimation algorithms. Relying on a known visual pattern, Mirzaei and Roumeliotis [96] showed one of the first online methods for extrinsic Inertial Measurement Unit (IMU)-camera calibration and IMU bias estimation. Later, Kelly and Sukhatme [79], Jones and Soatto [73], as well as Weiss et al. [139] presented different frameworks for visual-inertial navigation including the co-estimation of calibration parameters. All of these authors emphasize the importance of analyzing the observability characteristics of the underlying system and discuss the related issues. Recently, Leutenegger et al. [83] presented a delayed framework in which the authors included visual and inertial error terms into a extended nonlinear optimization in order to estimate the motion of a stereo camera as well as the landmarks in the map.

Efforts have also been done in order to find other visual error terms for combining the image information with inertial measurements. For example, Diel et al. [29] directly use the epipolar constraint between two matching features in subsequent frames as innovation term for their Kalman filter and thereby fuse the visual information with the accelerometer measurements (the gyroscopes and attitude are handled separately). By making the assumption that all features lie on a single plane, Omari et al. [107] derive a visual error term for optical flow measurements and combine it with inertial measurements by means of an UKF. Both approaches have in common that the 3D position of the features are not included into the state of the filter which significantly reduces the computational costs. Similarly, Mourikis and Roumeliotis [99] also exclude the position of the features from the states of their filter and introduce a measurement

model in order to account for the information when a feature is measured in multiple camera frames.

The primary goal of the present work is to propose a simple and reliable framework for the estimation of quantities which are critical for the safe operation of autonomous robots. We want to emphasize that we do not focus on achieving high-precision position and attitude accuracy, rather, our goal is to achieve a robust estimation of the velocity and inclination angle of the robot. This is especially important for systems which are controlled through dynamic motion, such as legged robots or quadrocopters. For this reason, we introduce visual error term which can directly extract information from a single feature match and does not rely on repeated measurements of the same feature. The above mentioned work of Diel et al. [29] is the closest to the present approach. In contrast to it, we propose the use of a different visual error term and co-estimate the inverse scene depth. By means of an UKF, we carry out a *tight* fusion of the visual and inertial measurements, whereby gyroscope and accelerometer measurement are included during the prediction step and the visual error terms serve as innovation during the update step. The presented approach is supported by a full nonlinear observability analysis and evaluated on data from real experiments.

The remainder of this paper is structured as follows: After introducing the most important notations and conventions in section 2, we describe the structure of the filter including the prediction and update steps in section 3. In section 4 we show and discuss the result of the nonlinear observability analysis. The experimental setup is described in section 5. Finally, we discuss the obtained results in section 6 and conclude with section 7.

2 Prerequisites

For better readability we give a short overview on the employed notations and conventions. The coordinates, expressed in a frame A, of a vector from a point P to a point Q are denoted by ${}_A\boldsymbol{r}_{PQ}$. If B is a second coordinate frame, then $\boldsymbol{C}_{BA}$ maps the coordinates expressed in A to the corresponding coordinates in B. The rotation between both frames is generally parametrized by the unit quaternion $\boldsymbol{q}_{BA}$, with the corresponding mapping $\boldsymbol{C} : \boldsymbol{q}_{BA} \mapsto \boldsymbol{C}_{BA}$. Throughout the paper, we add a subscript k to a quantity v, if we want to talk about its value at a time t_k, i.e., $v_k = v(t_k)$. Two coordinate frames are of interest: the world fixed coordinate frame W and the sensor frame B. For the sake of simplicity the following derivation assumes that the camera and the IMU coordinate frames are aligned with B.

We handle rotations as elements of $SO(3)$, where, together with the exponential and logarithm map, difference and derivatives are defined on $\mathbb{R}^3$. This is of high importance for the setup of the filter as well as for the corresponding observability analysis. Please note, that for this reason, also derivatives containing quaternions will be three dimensional in the corresponding directions, e.g. $\dot{\boldsymbol{q}} = -\boldsymbol{\omega} \in \mathbb{R}^3$. More information on this can be found in our previous work [11].

3 Filter Setup

3.1 Optical Flow and Visual Error Term

Based on the assumption of a static scene the following identity can be directly derived using kinematics relations only:

$$0 = {}_B\boldsymbol{v}_B + ({}_B\boldsymbol{w}_B^{\times}\boldsymbol{m}_i + \boldsymbol{u}_i)\lambda_i + \boldsymbol{m}_i\dot{\lambda}_i, \tag{9.1}$$

where ${}_B\boldsymbol{v}_B$ and ${}_B\boldsymbol{w}_B$ are the robot-centric velocity and rotational rate. The quantities $\boldsymbol{m}_i$, $\boldsymbol{u}_i$ and λ_i are related to the optical flow of a static feature i and represent the unit length bearing vector, the optical flow vector, and the depth of the feature. The challenge here is to find a way to properly extract information out of the equation without having to co-estimate the depth (and it's derivative) for each single optical flow measurement. A very common approach is to employ the continuous epipolar constraint which results from the above equation if left-multiplied by $\boldsymbol{m}_i^T({}_B\boldsymbol{w}_B^{\times}\boldsymbol{m}_i + \boldsymbol{u}_i)^{\times}$:

$$0 = \boldsymbol{m}_i^T({}_B\boldsymbol{w}_B^{\times}\boldsymbol{m}_i + \boldsymbol{u}_i)^{\times}\,{}_B\boldsymbol{v}_B. \tag{9.2}$$

This corresponds to an analytical elimination of the depth and its derivative. The problem is that this reduction does not consider the stochastic nature of the system and draws the estimation process towards singularities, e.g. zero velocity, which don't correspond to the maximum likelihood estimate (which is in general a desirable target for estimation). As a trade-off we propose to eliminate the derivative of the depth analytically by left-multiplying the equation by a 2×3 matrix $\boldsymbol{M}_i$ which fulfills:

$$\boldsymbol{M}_i\boldsymbol{m}_i = 0 \quad \wedge \quad \boldsymbol{M}_i\boldsymbol{M}_i^T = \boldsymbol{I}_2. \tag{9.3}$$

Additionally we make use of an inverse-depth parametrization, $\alpha_i = 1/\lambda_i$, and obtain

$$0 = \boldsymbol{M}_i\left({}_B\boldsymbol{v}_B\alpha_i + ({}_B\boldsymbol{w}_B^{\times}\boldsymbol{m}_i + \boldsymbol{u}_i)\right). \tag{9.4}$$

In comparison to the continuous epipolar constraint, this term retains more of the original constraint and is less susceptible to singularities. However, it also still contains one additional unknown, α_i, per visual feature. In order to cope with this, we will assume that the inverse depths α_i exhibit a Gaussian distribution around a mean α with standard deviation σ_α. The new parameter α corresponds to the inverse scene depth and will be co-estimated in the estimation process.

3.2 Filter States and Prediction Equations

The states of a filter have to be selected such that appropriate prediction and measurement equation can be derived. We define the following filter states:

$$\boldsymbol{x} := \left(\boldsymbol{r}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{c}, \boldsymbol{d}, \alpha\right), \tag{9.5}$$

$$:= \left({}_W\boldsymbol{r}_{WB}, {}_B\boldsymbol{v}_B, \boldsymbol{q}_{WB}, {}_B\boldsymbol{b}_f, {}_B\boldsymbol{b}_\omega, \alpha\right), \tag{9.6}$$

where $\boldsymbol{r}$ is the world position of the sensor, $\boldsymbol{v}$ represents its robot-centric velocity, $\boldsymbol{q}$ parametrizes the rotation between the sensor and the world coordinate frame, and $\boldsymbol{c}$ and $\boldsymbol{d}$ are the biases of the accelerometer and gyroscope. The additional state α is the inverse scene depth which is used for incorporating the optical flow measurements. The advantage of the robot-centric choice of states is that we thereby partition the state into non-observable states (absolute position and yaw) and observable states and thus avoid numerical problems related to non-observable states. A small drawback is that the noise of the gyroscope propagates onto the velocity state as well. Since, as will be shown later, the robot-centric velocity is fully observable, the additional noise can be compensated by the filter.

Analogous to other fusion algorithms including inertial measurements, we embed the proper acceleration measurement $\tilde{\boldsymbol{f}}$ and the rotational rate measurement $\tilde{\boldsymbol{\omega}}$ of the IMU directly into the prediction step of the proposed filter. Assuming that both measurements are affected by white Gaussian noise, $\boldsymbol{w}_f$ and $\boldsymbol{w}_\omega$, and additive bias terms, $\boldsymbol{c}$ and $\boldsymbol{d}$, we can write down

$$\tilde{\boldsymbol{f}} = \boldsymbol{f} + \boldsymbol{c} + \boldsymbol{w}_f, \tag{9.7}$$

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{d} + \boldsymbol{w}_\omega. \tag{9.8}$$

Both quantities are related to the kinematics of the sensor by

$$\boldsymbol{f} = \boldsymbol{C}(\boldsymbol{q}_{BW})\left({}_W\dot{\boldsymbol{v}}_B - \boldsymbol{g}\right), \tag{9.9}$$

$$\boldsymbol{\omega} = -\dot{\boldsymbol{q}}_{BW}, \tag{9.10}$$

where $\boldsymbol{g}$ is the gravity vector in W. By evaluating the total derivative of the filter states and combining it with the inertial measurements we obtain the following continuous time differential equations:

$$\dot{\boldsymbol{r}} = \boldsymbol{C}(\boldsymbol{q})\boldsymbol{v} + \boldsymbol{w}_r, \tag{9.11}$$

$$\dot{\boldsymbol{v}} = -\left(\tilde{\boldsymbol{\omega}} - \boldsymbol{d} - \boldsymbol{w}_\omega\right)^\times \boldsymbol{v} + \tilde{\boldsymbol{f}} - \boldsymbol{c} - \boldsymbol{w}_f + \boldsymbol{C}^T(\boldsymbol{q})\boldsymbol{g}, \tag{9.12}$$

$$\dot{\boldsymbol{q}} = \boldsymbol{C}(\boldsymbol{q})(\tilde{\boldsymbol{\omega}} - \boldsymbol{d} - \boldsymbol{w}_\omega), \tag{9.13}$$

$$\dot{\boldsymbol{c}} = \boldsymbol{w}_c, \tag{9.14}$$

$$\dot{\boldsymbol{d}} = \boldsymbol{w}_d, \tag{9.15}$$

$$\dot{\alpha} = w_\alpha. \tag{9.16}$$

The additional continuous white Gaussian noise processes $\boldsymbol{w}_c$ and $\boldsymbol{w}_d$ model a certain drift affecting the bias terms. w_α is included in order to handle varying inverse scene depths and $\boldsymbol{w}_r$ is included for being able to excite the full filter state and for modeling errors caused by the subsequent discretization of the states. For all white Gaussian noise processes, the corresponding covariance parameters, $\boldsymbol{R}_r$, $\boldsymbol{R}_f$, $\boldsymbol{R}_\omega$, $\boldsymbol{R}_c$, $\boldsymbol{R}_d$, and $\boldsymbol{R}_\alpha$ describe the magnitude of the noise. Except for $\boldsymbol{R}_r$ and $\boldsymbol{R}_\alpha$ which are tuning parameters, all covariance parameters can be identified by considering the Allan plots of the IMU measurements [32].

The discretization is based on a simple Euler forward integration scheme. Please note that for the rotational states, the step forward can be taken on the corresponding

sigma algebra and then be mapped back onto $SO(3)$. This corresponds to:

$$\boldsymbol{q}(t_k) = \exp\left(\Delta t_k \dot{\boldsymbol{q}}(t_{k-1})\right) \otimes \boldsymbol{q}(t_{k-1}), \tag{9.17}$$

with

$$\Delta t_k = t_k - t_{k-1}. \tag{9.18}$$

This leads to:

$$\boldsymbol{r}_k = \boldsymbol{r}_{k-1} + \Delta t_k \left(\boldsymbol{C}_{k-1}\boldsymbol{v}_{k-1} + \boldsymbol{w}_{r,k}\right), \tag{9.19}$$

$$\begin{aligned}\boldsymbol{v}_k = &\left(I - \Delta t_k \left(\tilde{\boldsymbol{\omega}}_k - \boldsymbol{d}_{k-1} - \boldsymbol{w}_{\omega,k}\right)^{\times}\right)\boldsymbol{v}_{k-1} \\ &+ \Delta t_k \Big(\tilde{\boldsymbol{f}}_k - \boldsymbol{c}_{k-1} - \boldsymbol{w}_{f,k} + \boldsymbol{C}_{k-1}^T \boldsymbol{g}\Big),\end{aligned} \tag{9.20}$$

$$\boldsymbol{q}_k = \exp\Big(\Delta t_k \boldsymbol{C}_{k-1}(\tilde{\boldsymbol{\omega}}_k - \boldsymbol{d}_{k-1} - \boldsymbol{w}_{\omega,k})\Big) \otimes \boldsymbol{q}_{k-1}, \tag{9.21}$$

$$\boldsymbol{c}_k = \boldsymbol{c}_{k-1} + \Delta t_k \boldsymbol{w}_{c,k}, \tag{9.22}$$

$$\boldsymbol{d}_k = \boldsymbol{d}_{k-1} + \Delta t_k \boldsymbol{w}_{d,k}, \tag{9.23}$$

$$\alpha_k = \alpha_{k-1} + \Delta t_k w_{\alpha,k}. \tag{9.24}$$

3.3 Measurement Equations

The measurement equations are directly based on the findings of section 3.1. For each available optical flow measurement i we directly define the corresponding 2D innovation term for the filter:

$$\boldsymbol{y}_i = \boldsymbol{M}_i \left(\boldsymbol{v}\ \alpha_i + (\boldsymbol{\omega}^{\times}\boldsymbol{m}_i + \boldsymbol{u}_i)\right). \tag{9.25}$$

As discussed above, we introduced the inverse scene depth as a filter state and thus model deviations of the single inverse depths α_i as measurement noise:

$$\alpha_i = \alpha + n_{\alpha,i}, \quad n_{\alpha,i} \sim \mathcal{N}(0, \sigma_\alpha^2). \tag{9.26}$$

Furthermore, we also have to model noise on the bearing vectors $\boldsymbol{m}_i$ and optical flow vectors $\boldsymbol{u}_i$. For typical scenarios the major part of the uncertainties originate through $\boldsymbol{u}_i$, which lies in the orthogonal subspace of $\boldsymbol{m}_i$. Thus, we can introduce an additive lumped noise term on $\boldsymbol{u}_i$, whereby it is sufficient to excite directions orthogonal to $\boldsymbol{m}_i$ only. This can be achieved by means of the previously defined matrix $\boldsymbol{M}_i$ ($\boldsymbol{n}_u$ is two dimensional):

$$\tilde{\boldsymbol{u}}_i = \boldsymbol{u}_i - \boldsymbol{M}_i^T \boldsymbol{n}_u, \tag{9.27}$$

$$\boldsymbol{n}_u \sim (0, \boldsymbol{R}_u). \tag{9.28}$$

With this the innovation term becomes:

$$\boldsymbol{y}_i = \boldsymbol{M}_i \left(\boldsymbol{v}(\alpha + n_{\alpha,i}) + (\boldsymbol{\omega}^{\times}\boldsymbol{m}_i + \tilde{\boldsymbol{u}}_i)\right) + \boldsymbol{n}_u. \tag{9.29}$$

The parameter $\boldsymbol{R}_u$ describes the accuracy of the visual measurements and the parameter σ_α^2 depends on the variance of the inverse depths in the scene.

An interesting effect is that whenever the velocity is small or when the inverse scene depth tends towards zero (i.e. the scene is far away), the innovation term will be equivalent to a visual gyroscope:

$$\boldsymbol{y}_i^* = \boldsymbol{M}_i \left((\boldsymbol{\omega}^\times \boldsymbol{m}_i + \tilde{\boldsymbol{u}}_i) \right) + \boldsymbol{n}_u. \tag{9.30}$$

3.4 Unscented Kalman Filter and Outliers Detection

An unscented Kalman filter (UKF) is employed as filtering framework. The main reason for this is that the UKF can handle correlated noise between prediction and update by using a single set of augmented sigma points for both steps. All equations required for its implementation are the prediction equation (9.19)-(9.24) and the update equation (9.30), whereby the single innovation terms of the multiple features are stuck together. The twofold use of the gyroscope measurement can be directly seen in these equations. Please note that the implementation has to take into account that, although the attitude is parametrized by a unit quaternion, the corresponding noise and perturbations are always on a 3D subspace. For a detailed discussion on the employed UKF itself please refer to [74].

In order to handle the high sensitivity of Kalman filters to outliers, we implement a simple outliers detection method on the innovation terms. Using an analogous approach as Mirzaei et al. [96], we reject a visual measurement whenever the Mahalanobis distance of the corresponding innovation terms exceeds a certain threshold. The predicted covariance of the innovation is used as weighting for the Mahalanobis distance and the threshold is chosen in such a manner that, in theory, 5% of the inliers are rejected. Considering that the underlying probability distribution is a χ^2-distribution with two degrees of freedom the threshold is set to $p = 5.99$. In summary, the criteria for rejecting a measurement i is given by (where $\boldsymbol{S}_i$ is the predicted covariance matrix):

$$\boldsymbol{y}_i^T \boldsymbol{S}_i^{-1} \boldsymbol{y}_i > p. \tag{9.31}$$

4 Observability Analysis

A nonlinear observability analysis is carried out for the proposed system. A detailed discussion of the theory behind it was provided by Hermann and Krener [57]. In the scope of this paper we only outline the rough procedure of the analysis. Based on the nonlinear representation of the system an observability matrix is derived in order to assess the observability characteristics of the system. The system can be written as follows, whereby the noise quantities can be ignored since they don't affect the

observability analysis:

$$\dot{\boldsymbol{x}} = \begin{pmatrix} \boldsymbol{C}\boldsymbol{v} \\ \hat{\boldsymbol{\omega}}^{\times}\boldsymbol{v} - \hat{\boldsymbol{f}} + \boldsymbol{C}^T\boldsymbol{g} \\ -\boldsymbol{C}\hat{\boldsymbol{\omega}} \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{9.32}$$

$$\boldsymbol{h}_i(\boldsymbol{x}) = \boldsymbol{M}_i \left(\boldsymbol{v}\ \alpha + (-\hat{\boldsymbol{w}}^{\times}\boldsymbol{m}_i + \tilde{\boldsymbol{u}}_i) \right), \tag{9.33}$$

with the shortcuts

$$\hat{\boldsymbol{f}} = -\ \tilde{\boldsymbol{f}} + \boldsymbol{c}, \tag{9.34}$$

$$\hat{\boldsymbol{\omega}} = -\ \tilde{\boldsymbol{\omega}} + \boldsymbol{d}. \tag{9.35}$$

The observability matrix is composed of the gradient of the Lie derivatives of the above system, whereby $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{\omega}}$ are, in the context of this analysis, the inputs to the system. We can show, that if there are three optical measurements with non-coplanar bearing vectors and if the inverse scene depth is not zero we can simplify the observability matrix to the following term (if $\alpha = 0$ only the gyroscope bias and the inverse scene depth itself (if $\boldsymbol{v} \neq 0$) are observable):

$$\boldsymbol{O} = \begin{bmatrix} 0 & \boldsymbol{I} & 0 & 0 & 0 & \frac{1}{\alpha}\boldsymbol{v} \\ 0 & 0 & 0 & 0 & \boldsymbol{I} & 0 \\ 0 & 0 & \boldsymbol{C}^T\boldsymbol{g}^{\times} & -\boldsymbol{I} & 0 & \boldsymbol{C}^T\boldsymbol{g} - \hat{\boldsymbol{f}} \\ 0 & 0 & \hat{\boldsymbol{\omega}}^{\times}\boldsymbol{C}^T\boldsymbol{g}^{\times} & 0 & 0 & \hat{\boldsymbol{\omega}}^{\times}\boldsymbol{C}^T\boldsymbol{g} \end{bmatrix}. \tag{9.36}$$

Throughout the analysis only rank-preserving row operations are carried out which keeps the relation between each column and a specific state of the filter. We also have to keep in mind, that $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{\omega}}$ represent system inputs in this analysis, and thus a single line in the matrix can be duplicated by inserting different values for $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{\omega}}$ (see [57]). By inserting two non-colinear values for $\tilde{\boldsymbol{\omega}}$ (through $\hat{\boldsymbol{\omega}}$) in the last row of the matrix we can further simplify the matrix to:

$$\boldsymbol{O} = \begin{bmatrix} 0 & \boldsymbol{I} & 0 & 0 & 0 & \frac{1}{\alpha}\boldsymbol{v} \\ 0 & 0 & 0 & 0 & \boldsymbol{I} & 0 \\ 0 & 0 & 0 & -\boldsymbol{I} & 0 & -\hat{\boldsymbol{f}} \\ 0 & 0 & \boldsymbol{C}^T\boldsymbol{g}^{\times} & 0 & 0 & \boldsymbol{C}^T\boldsymbol{g} \end{bmatrix}. \tag{9.37}$$

The rank of this matrix is 12 (independent of the choice of $\boldsymbol{C}$, $\boldsymbol{v}$, or $\hat{\boldsymbol{f}}$) and the dimension of the right null-space is consequently 4, which is spanned by the following matrix:

$$\boldsymbol{N} = \begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & 0 \\ 0 & \boldsymbol{g} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{9.38}$$

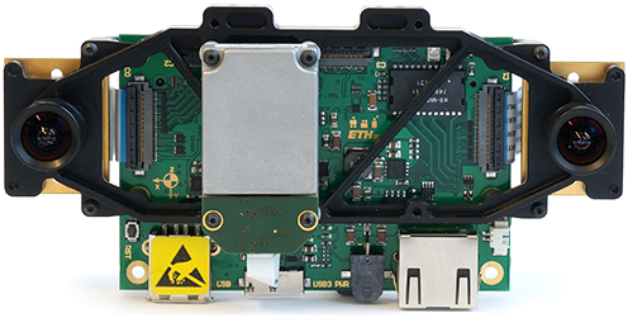

Figure 9.1: ASL visual-inertial SLAM sensor employed for evaluating the presented optical flow and inertial measurement fusion approach.

In an informal way, the perturbations along the directions spanned by $\boldsymbol{N}$ cannot be perceived at the filter output. While the first column corresponds to the absolute position of the system, the second column represents a rotation around the gravity axis, i.e., global position and yaw angle are not observable. Mathematically this can be written as:

$$\boldsymbol{r}^* = \boldsymbol{r} + \delta\boldsymbol{r}, \tag{9.39}$$

$$\boldsymbol{q}^* = \exp(\boldsymbol{g}\delta\psi) \otimes \boldsymbol{q}, \tag{9.40}$$

where $\delta\boldsymbol{r}$ and $\delta\psi$ are perturbations. $\boldsymbol{r}$ and $\boldsymbol{q}$ cannot be distinguished from $\boldsymbol{r}^*$ and $\boldsymbol{q}^*$, respectively.

All in all, the above nonlinear observability analysis allows us to state that for all points in the state-space (except if $\alpha = 0$) there *exists* some input $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{\omega}}$ (corresponding to a certain motion of the sensor) such that all states are locally weakly observable, except for the global position and yaw angle.

5 Experimental Setup

To validate the proposed scheme, the Unscented Kalman filter was implemented in C++. The filter was tested on data that were recorded using the ASL visual-inertial SLAM sensor (see fig. 9.1), with synchronized global-shutter camera (Aptina MT9-V034 at 20 Hz) and IMU (Analog Devices ADIS16488 at 200 Hz). The pose of the sensor was additionally tracked using a Vicon motion tracking system at 100 Hz.

The image features are tracked using a Lukas-Kanade-based tracker. Salient image features that are used for tracking are extracted by first applying a FAST corner detector, computing the Shi-Tomasi score for each extracted corner and then selecting those corners which have the highest score while ensuring a uniform distribution of

the features in the image. A uniform feature distribution is ensured by masking parts of the images that are already populated with strong features and by only adding new, weaker features in unpopulated image regions.

Feature extraction and LK-tracking for 150 features is taking less than 2.5 ms in total on a single core of an Intel i7-3740QM processor for one frame. Equivalently, a measurement update step using 50 optical flow features is performed in 10 ms. During the experiments an average feature count of 50 features was used. The rather bad scalability of the filter update can be easily overcome by changing to the information form of the filter, which will be part of future work.

6 Results and Discussion

The presented approach was evaluated on different datasets from an indoor environment where the feature depths range between 0.5 m and 5 m. The motion of the sensor included rotational rate of up to 3 rad/s. Our main goal was to develop a filter for delivering high-rate and reliable state estimates rather than being mainly focused on estimation accuracy. Furthermore, the main states of interest are the velocities and the inclination angles since they are of major importance if it comes to control of dynamic robot motions. Using a 2 minute long dataset where the sensor was excited along its different degrees of freedom, the following RMS values where obtained:

- Attitude (rad): 0.027 (roll), 0.005 (pitch) , 0.074 (yaw)
- Velocity (m/s): 0.058 (x), 0.070 (y), 0.075 (z)

whereby the velocity is always evaluated in the sensor frame B. When using different datasets with similar motions the RMS values fluctuate around the above values, except for the RMS of the yaw angle which increases with the estimation time (since it is not observable). The estimated IMU biases converge relatively fast depending on the motion of the system. While we have no ground truth values for the bias terms, figure 9.2 and 9.3 show the typical convergence of the biases when the system is being excited along its different directions. Figuring out which direction needs to be excited for improving the estimation of a certain state can be a very difficult problem and is not within the scope of this paper. The 3σ-bounds of the covariance matrix are plotted as dashed lines.

Figure 9.4, 9.5, and 9.6 present the results from a dataset where after some initial motion the sensor holds still for awhile before being moved again. This can be clearly seen between $33 - 43$ seconds. In contrast to the standard epipolar constraint, the employed visual error term still extracts information from the optical flow measurements analogous to a visual gyroscope. Still, during this phase additional uncertainty accumulates in the different states. However, as soon as the sensor is moved again, the observable states very quickly converge back to the reference. This can be nicely observed for the velocity estimates. Note as well, that although the position of the sensor is unobservable, it can be corrected and loose uncertainty to some extent through the cross-correlation it maintains with the other states.

When replacing the presented visual error term by other terms such as the simple continuous epipolar constraint or normalized forms of it, the observed results became

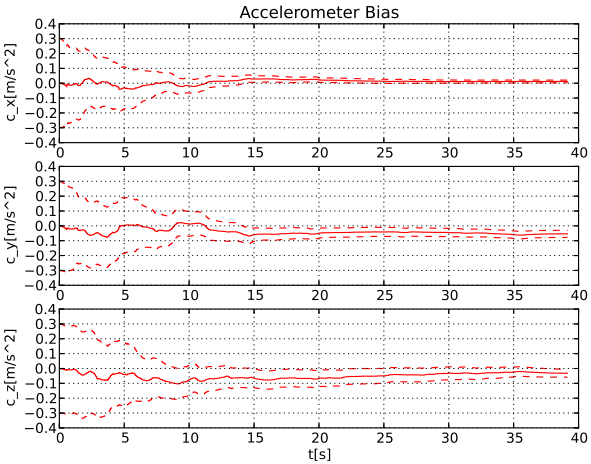

Figure 9.2: Estimated accelerometer biases. Red: estimated values. Red dashed line: 3σ-bound. The initial converges is supported by motion of the sensor. The estimate is more accurate along the x-axis because it is more often aligned with the gravity axis.

worse. Very often the estimation process would be drawn to zero (e.g. for the continuous epipolar constraint) or very quickly lead to bad tracking or divergence.

All in all the filter exhibits a rather average performance in terms of accuracy when compared with the state of the art visual-inertial algorithms. However, when considering that only frame to frame (20 Hz) information is included into the filter, the obtained results are relatively surprising, especially since other quantities like the IMU biases have to be co-estimated simultaneously. A major advantage of this approach is that the filter is free of any complex initialization procedure and only relies on single feature matches between subsequent frames. It does not require the long term tracking of some feature and is thus much less affected by fast motions.

7 Conclusion and Future Work

In this paper we presented a relative simple approach for fusing optical flow and inertial measurements. By deriving a special optical flow error term and embedding it into an UKF framework, we were able to derive a filter for estimating the egomotion of the sensor, the IMU biases as well as the inverse scene depth. By carrying out a nonlinear observability analysis we showed that all states except for the global position and yaw angle are locally weakly observable. The results obtained on a real dataset

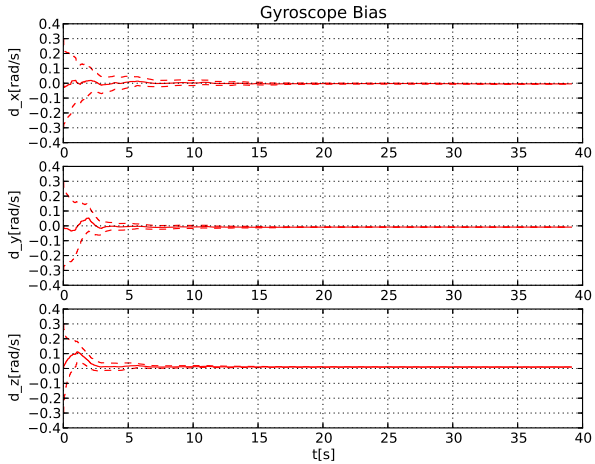

Figure 9.3: Estimated gyroscope biases. Red: estimated values. Red dashed line: 3σ-bound. The states here converge faster than the accelerometer biases since the optical flow measurement have a direct impact on the angular rates.

confirmed that the filter was able to estimate the different observable states.

One important aspect of future work will be the combination of the presented approach with other visual localization methods. While the strength of the presented approach lies in its robustness and speed, it could be combined together with some static feature tracking in order to improve its accuracy and long term stability. Other possible extensions include the implementation on multiple cameras or the combination with further sensor modalities.

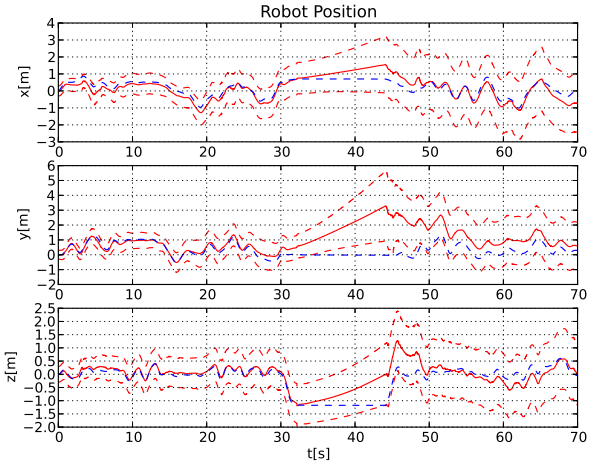

Figure 9.4: Estimated sensor position. Red: estimated values. Red dashed line: 3σ-bound. Dashed blue line: motion capture ground truth. The position state is affected by increasing uncertainty since it is not observable and represents the integration of the velocity estimate.

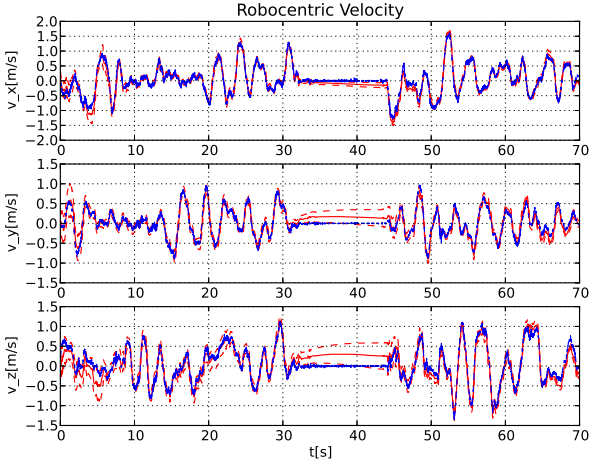

Figure 9.5: Estimated sensor velocity expressed in the sensor coordinate frame itself. Red: estimated values. Red dashed line: 3σ-bound. Dashed blue line: motion capture ground truth. The robot-centric velocity is fully observable and consequently has a bounded uncertainty. Even after a phase of increased uncertainty it is able to recover if sufficient excitation is available.

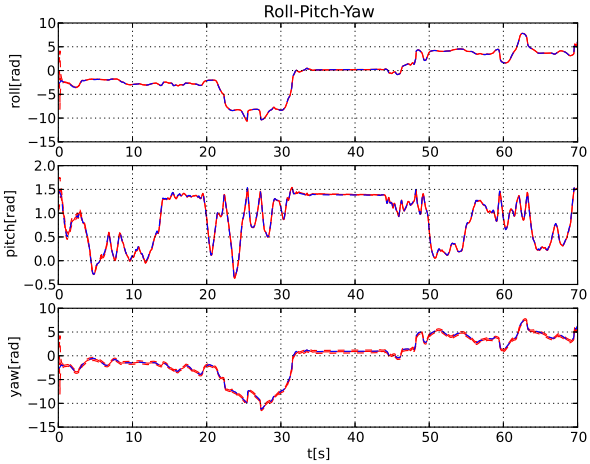

Figure 9.6: Roll, pitch, and yaw angle of the sensor. Red: estimated values. Red dashed line: 3σ-bound. Dashed blue line: motion capture ground truth. Pitch and roll are observable and consequently exhibit a nice tracking behavior. Yaw is not observable and slowly drifts away.

Paper VI

Robust Visual Inertial Odometry Using a Direct EKF-Based Approach

Michael Bloesch, Sammy Omari, Marco Hutter, Roland Siegwart

Abstract

In this paper, we present a monocular visual-inertial odometry algorithm which, by directly using pixel intensity errors of image patches, achieves accurate tracking performance while exhibiting a very high level of robustness. After detection, the tracking of the multilevel patch features is closely coupled to the underlying extended Kalman filter (EKF) by directly using the intensity errors as innovation term during the update step. We follow a purely robocentric approach where the location of 3D landmarks are always estimated with respect to the current camera pose. Furthermore, we decompose landmark positions into a bearing vector and a distance parametrization whereby we employ a minimal representation of differences on a corresponding Lie-Algebra in order to achieve better consistency and to improve the computational performance. Due to the robocentric, inverse-distance landmark parametrization, the framework does not require any initialization procedure, leading to a truly power-up-and-go state estimation system. The presented approach is successfully evaluated in a set of highly dynamic hand-held experiments as well as directly employed in the control loop of a multirotor unmanned aerial vehicle (UAV).

Published in:
IEEE/RSJ International Conference on Intelligent Robots and Systems, 2015
DOI: 10.1109/IROS.2015.7353389

1 Introduction

Navigation and control of autonomous robots in rough and highly unstructured environments requires high-bandwidth and precise knowledge of position and orientation. Especially in dynamic operation of robots, the underlying state estimation can quickly become the bottleneck in terms of achievable bandwidth, robustness and speed. To enable the required performance for highly dynamic operation of robots, we combine complementary information from vision- and inertial sensors. This approach has a long history and has been successfully applied to navigate unmanned aerial robots [139], [119], walking robots [128], [141] or cars [44].

Within the field of computer vision, Davison et al. [28] proposed one of the first real-time 3D monocular localization and mapping frameworks. Since then, a lot of improvements have been contributed from various research groups and further approaches have been proposed. A key issue is to improve the consistency of the estimation framework which is affected by its inherent nonlinearity [21, 75]. One approach is to make use of a robocentric representation for the tracked features and thereby significantly reduce the effect of nonlinearities [21, 25]. As alternative, Huang et al. [61] propose the use of a so-called observability constrained extended Kalman filter, whereby the inconsistencies can be avoided by using special linearization points while evaluating the system Jacobians.

A somewhat related problem is the choice of the specific representation of the features. Since for monocular setups, the depth of a newly detected feature is unknown the initial 3D location estimate of the feature exhibits a high (infinite) uncertainty along the corresponding axis. In order to integrate this feature from the beginning into the estimation framework, Montiel et al. [98] proposed the use of an inverse-depth parametrization (IDP). With this parametrization, each feature location is represented by the camera position where the feature was initially detected, by a bearing vector (parametrized with azimuth and elevation angle), as well as the inverse depth of the feature. The resulting increase in consistency was analyzed in more detail for the IDP and other feature parametrization in [126].

While most standard visual odometry approaches are based on detected and tracked point features as source of visual information, so-called *direct* approaches directly use the image intensities in their estimation framework. Especially with the recent advent of RGBD cameras, so called *dense* approaches, where the intensity error over the full image is considered, have gained a lot of attention [3, 80]. In comparison to traditional vision-based state estimators, dense approaches have a significantly larger error term count and require appropriate methods in order to tackle the additional computational load. By employing highly optimized SSE-vectorized implementations, first real-time, CPU-based approaches for dense- or semi-dense motion estimation using a RGBD [80] or a monocular RGB camera [33, 40] have recently been proposed.

Incorporating inertial measurements in the estimation can significantly improve the robustness of the system, provides the estimation process with the notion of gravity, and allows for a more accurate and high bandwidth estimation of the velocities and rotational rates. By adapting the original EKF proposed by Davison et al. [28], additional IMU measurements can be relatively simply integrated into the ego-motion estimation, whereby calibration parameters can be co-estimated online [73, 79]. Leutenegger et al. [83] describe a *tightly* coupled approach in which the robot trajectory

and sparse 3D landmarks are estimated in a joint optimization problem using inertial error terms as well as the reprojection error of the landmarks in the camera image. This is done in a windowed bundle adjustment approach over a set of keyframe images and a temporal inertial measurement window. Similarly, in [99] the authors estimate the trajectory in an IMU-driven filtering framework using the reprojection error of 3D landmarks as measurement updates. Instead of adding the landmarks to the filter state, they immediately marginalize them out using a nullspace decomposition, thus leading to a small filter state size.

In the present paper we propose a visual-inertial odometry framework which combines and extends several of the above mentioned approaches. While targeting a simple and consistent approach and avoiding ad-hoc solutions, we adapt the structure of the standard visual-inertial EKF-SLAM formulation [73, 79]. The following keypoints are integrated into the proposed framework:

- Point features are parametrized by a bearing vector and a distance parameter with respect to the current frame. A suitable σ-Algebra is used for deriving the corresponding dynamics and performing filtering operations.
- Multilevel patch features are directly tracked within the EKF, whereby the intensity errors are used as innovation terms during the update step.
- A QR-decomposition is employed in order to reduce the high dimensional error terms and thus keep the Kalman update computationally tractable.
- A purely robocentric representation of the *full* filter state is employed. The camera extrinsics as well as the additive IMU biases are also co-estimated.

Together this yields a *fully robocentric* and *direct* monocular visual-inertial odometry framework which can be run real-time on a single standard CPU core. In several experiments on real data we show its reliable and accurate tracking performance while exhibiting a high robustness against fast motions and various disturbances. The framework is implemented in c++ and is available as open-source software [15].

2 Filter Setup

2.1 Overall Filter Structure and State Parametrization

The overall structure of the filter is derived from the one employed in [73, 79]: The inertial measurements are used to propagate the state of the filter, while the visual information is taken into account during the filter update steps. As a fundamental difference we make use of a fully robocentric representation of the filter state which can be seen as an adaptation of [25] (which is vision-only). One advantage of this formulation is that problems with unobservable states can inherently be avoided and thus the consistency of the estimates can be improved. On the other hand noise from the gyroscope will affect all states that need to be rotated during the state propagation (see section 2.2). However, since the gyroscope noise is relatively small and because most states are observable this does not represent a significant issue.

Three different coordinate frames are used throughout the paper: the inertial world coordinate frame, $\mathcal{I}$, the IMU fixed coordinate frame, $\mathcal{B}$, as well as the camera fixed coordinate frame, $\mathcal{V}$. For tracking N visual features, we use the following filter state:

$$\boldsymbol{x} := \left(\boldsymbol{r}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{b}_f, \boldsymbol{b}_\omega, \boldsymbol{c}, \boldsymbol{z}, \boldsymbol{\mu}_0, \ldots, \boldsymbol{\mu}_N, \rho_0, \ldots, \rho_N\right), \tag{10.1}$$

with:

- $\boldsymbol{r}$: robocentric position of IMU (expressed in $\mathcal{B}$),
- $\boldsymbol{v}$: robocentric velocity of IMU (expressed in $\mathcal{B}$),
- $\boldsymbol{q}$: attitude of IMU (map from $\mathcal{B}$ to $\mathcal{I}$),
- $\boldsymbol{b}_f$: additive bias on accelerometer (expressed in $\mathcal{B}$),
- $\boldsymbol{b}_\omega$: additive bias on gyroscope (expressed in $\mathcal{B}$),
- $\boldsymbol{c}$: translational part of IMU-camera extrinsics (expressed in $\mathcal{B}$),
- $\boldsymbol{z}$: rotational part of IMU-camera extrinsics (map from $\mathcal{B}$ to $\mathcal{V}$),
- $\boldsymbol{\mu}_i$: bearing vector to feature i (expressed in $\mathcal{V}$),
- ρ_i: distance parameter of feature i.

The generic parametrization for the distance d_i of a feature i is given by the mapping $d_i = d(\rho_i)$ (with derivative $d'(\rho_i)$). In the context of this work we mainly tested the inverse distance parametrization, $d(\rho_i) = 1/\rho_i$. The investigation of further parametrization will be part of future work.

Rotations ($\boldsymbol{q}, \boldsymbol{z} \in SO(3)$) and unit vectors ($\boldsymbol{\mu}_i \in S^2$) are parametrized by following the approach of Hertzberg et al. [59]. This is required in order to perform operations like computing differences or derivatives as well as representing the uncertainty of the state in a minimal manner. For parametrizing unit vectors we employ rotations as underlying representation, whereby we define a $\boxminus$-operator which returns a difference between two unit vectors within a 2D linear subspace. The advantage of this parametrization is that the tangent space can be easily computed (which is used for defining the $\boxminus$-operator).

By using the combined bearing vector and distance parameterization, features can be initialized in an *undelayed* manner, i.e., the features are integrated into the filter at detection. The distance of a feature is initialized with a fixed value or, if sufficiently converged, with an estimate of the current average scene distance. The corresponding covariance is set to a very large value. In comparison to other parameterizations we do not over-parametrize the 3D feature location estimates, whereby each feature corresponds to 3 columns in the covariance matrix of the state (2 for the bearing vector and 1 for the distance parameter). This also avoids the need for re-parameterization [126].

2.2 State Propagation

Based on the proper acceleration measurement, $\tilde{\boldsymbol{f}}$, and the rotational rate measurement, $\tilde{\boldsymbol{\omega}}$, the evaluation of the IMU driven state propagation results in the following set of continuous differential equations (the superscript $^\times$ denotes the skew symmetric matrix of a vector):

$$\dot{\boldsymbol{r}} = -\hat{\boldsymbol{\omega}}^\times \boldsymbol{r} + \boldsymbol{v} + \boldsymbol{w}_r, \tag{10.2}$$

$$\dot{\boldsymbol{v}} = -\hat{\boldsymbol{\omega}}^\times \boldsymbol{v} + \hat{\boldsymbol{f}} + \boldsymbol{q}^{-1}(\boldsymbol{g}), \tag{10.3}$$

$$\dot{\boldsymbol{q}} = -\boldsymbol{q}(\hat{\boldsymbol{\omega}}), \tag{10.4}$$

$$\dot{\boldsymbol{b}}_f = \boldsymbol{w}_{bf}, \tag{10.5}$$

$$\dot{\boldsymbol{b}}_\omega = \boldsymbol{w}_{b\omega}, \tag{10.6}$$

$$\dot{\boldsymbol{c}} = \boldsymbol{w}_c, \tag{10.7}$$

$$\dot{\boldsymbol{z}} = \boldsymbol{w}_z, \tag{10.8}$$

$$\dot{\boldsymbol{\mu}}_i = \boldsymbol{N}^T(\boldsymbol{\mu}_i)\hat{\boldsymbol{\omega}}_\mathcal{V} - \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \boldsymbol{N}^T(\boldsymbol{\mu}_i)\frac{\hat{\boldsymbol{v}}_\mathcal{V}}{d(\rho_i)} + \boldsymbol{w}_{\mu,i}, \tag{10.9}$$

$$\dot{\rho}_i = -\boldsymbol{\mu}_i^T \hat{\boldsymbol{v}}_\mathcal{V}/d'(\rho_i) + w_{\rho,i}, \tag{10.10}$$

where $\boldsymbol{N}^T(\boldsymbol{\mu})$ linearly projects a 3D vector onto the 2D tangent space around the bearing vector $\boldsymbol{\mu}$, with the bias corrected and noise affected IMU measurements:

$$\hat{\boldsymbol{f}} = \tilde{\boldsymbol{f}} - \boldsymbol{b}_f - \boldsymbol{w}_f, \tag{10.11}$$

$$\hat{\boldsymbol{\omega}} = \tilde{\boldsymbol{\omega}} - \boldsymbol{b}_\omega - \boldsymbol{w}_\omega, \tag{10.12}$$

and with the camera linear velocity and rotational rate:

$$\hat{\boldsymbol{v}}_\mathcal{V} = \boldsymbol{z}(\boldsymbol{v} + \hat{\boldsymbol{\omega}}^\times \boldsymbol{c}), \tag{10.13}$$

$$\hat{\boldsymbol{\omega}}_\mathcal{V} = \boldsymbol{z}(\hat{\boldsymbol{\omega}}). \tag{10.14}$$

Furthermore, $\boldsymbol{g}$ is the gravity vector expressed in the world coordinate frame, and the terms of the form $\boldsymbol{w}_*$ are white Gaussian noise processes. The corresponding covariance parameters can either be taken from the IMU specifications or have to be tuned manually. Using an appropriate Euler forward integration scheme, i.e., using the $\boxplus$-operator where appropriate, the above time continuous equation can be transformed into a set of discrete prediction equations which are used during the prediction of the filter state [59].

Please note that the derivatives of bearing vectors and rotations lie within 2D and 3D vector spaces, respectively. This is required for achieving a minimal and consistent representation of the filter state and covariance.

2.3 Filter Update

For every captured image we perform a state update. We assume that we know the intrinsic calibration of the camera and can therefore compute the projection of

a bearing $\boldsymbol{\mu}$ to the corresponding pixel coordinate $\boldsymbol{p} = \boldsymbol{\pi}(\boldsymbol{\mu})$. As will be described in section 3.2, we derive a 2D linear constraint, $\boldsymbol{b}_i(\boldsymbol{\pi}(\hat{\boldsymbol{\mu}}_i))$, for each feature i which is predicted to be visible in the current frame with bearing vector $\hat{\boldsymbol{\mu}}_i$. This linear constraint encompasses the intensity errors associated with a specific feature and can be directly employed as innovation term within the Kalman update (affected by additive discrete Gaussian pixel intensity noise $\boldsymbol{n}_i$):

$$\boldsymbol{y}_i = \boldsymbol{b}_i(\boldsymbol{\pi}(\hat{\boldsymbol{\mu}}_i)) + \boldsymbol{n}_i, \tag{10.15}$$

together with the Jacobian:

$$\boldsymbol{H}_i = \boldsymbol{A}_i(\boldsymbol{\pi}(\hat{\boldsymbol{\mu}}_i))\frac{d\boldsymbol{\pi}}{d\boldsymbol{\mu}}(\hat{\boldsymbol{\mu}}_i). \tag{10.16}$$

By stacking the above terms for all visible features we can directly perform a standard EKF update. However, if the initial guess for a certain bearing vector $\hat{\boldsymbol{\mu}}_i$ has a large uncertainty the update will potentially fail. This typically occurs if features get newly initialized and exhibit a large distance uncertainty. In order to avoid this issue we improve the initial guess for a bearing vector with large uncertainty by performing a patch based search of the feature (section 3.2). This basically improves the linearization point of the EKF by using the bearing vector obtained from the patch search $\bar{\boldsymbol{\mu}}_i$ for evaluating the terms in eqs. (10.15) and (10.16). Please note that the EKF update equations have to be slightly adapted in order to account for the altered linearization point. A similar alternative would be to directly employ an iterative EKF.

In order to account for moving objects or other disturbances, a simple Mahalanobis based outlier detection is implemented within the update step. It compares the obtained innovation with the predicted innovation covariance and rejects the measurement whenever the weighted norm exceeds a certain threshold. This method inherently takes into account the covariance of the state and measurements. For instance it also considers the image gradients and thereby tends to reject gradient-less image patches easier.

3 Multilevel Patch Feature Handling

Along the lines of other visual-inertial EKF approaches ([73, 79]) we fully integrate visual features into the state of the Kalman filter (see also section 2.1). Within the prediction step the new locations of the multilevel patch features are estimated by considering the IMU-driven motion model (eq. (10.9)). Especially if the calibration of the extrinsics and the feature distance parameters have converged, this yields high quality predictions for the feature locations. Additionally, the covariance of the predicted pixel location can be easily computed and the computational effort of a possible pre-alignment strategy can be adapted accordingly. The subsequent update step computes an innovation term by evaluating the discrepancy between the projection of the multilevel patch into the image frame and the image itself. Considering the cross-correlation between the states the EKF spreads the resulting corrections throughout the filter state. In the following the different steps and algorithms involving feature handling are discussed in more details. The overall workflow for a single feature is depicted in fig. 10.1.

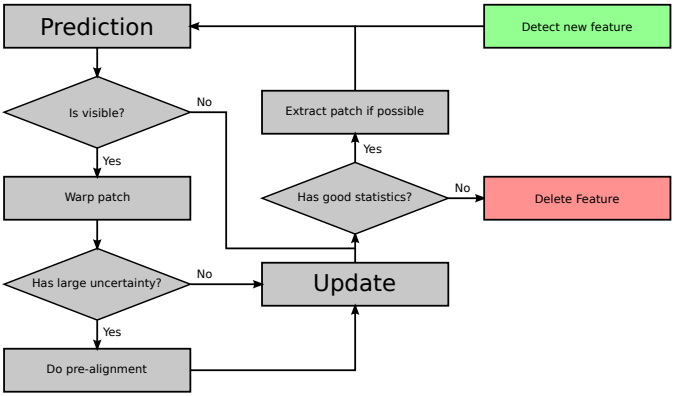

Figure 10.1: Overview on the workflow of a feature in the filter state. The heuristics for adding and removing features are adapted to the total number of possible features.

3.1 Structure and Warping

For a given image pyramid (factor 2 down-sampling) and a given bearing vector $\boldsymbol{\mu}$ a multilevel patch is obtained by extracting constant size (here 8x8 pixels) patches, P_l, for each image level l at the corresponding pixel coordinate $\boldsymbol{p} = \boldsymbol{\pi}(\boldsymbol{\mu})$. The advantage is that tracking such features is robust against bad initial guesses and image blur. Furthermore such patch features allow a *direct* intensity error feedback into the filter. In comparison to reprojection error based algorithms this allows to formulate a more accurate error model which inherently takes into account the texture of the tracked image patch. For instance it also enables the use of edge features, whereby the gained information would be along the perpendicular to the edge.

By tracking two additional bearing vectors within the patch, we can compute an affine warping matrix $\boldsymbol{W} \in \mathbb{R}^{2\times 2}$ in order to account for the local distortion of the patches between subsequent images. We assume that the distance of the feature is large w.r.t. the size of the patch and can thus choose the normal of the patches to point towards the center of the camera. Also, when a feature was successfully tracked within a frame, the multilevel patch is re-extracted in order to avoid the accumulation of errors.

3.2 Alignment Equations and QR-decomposition

Throughout the framework we make use of intensity errors in order to pre-align features or update the filter state. For a given image pyramid with images I_l and a given multilevel patch feature (with coordinates $\boldsymbol{p}$ and patches P_l) the following intensity

errors can be evaluated for image level l and patch pixel $\boldsymbol{p}_j$:

$$e_{l,j} = P_l(\boldsymbol{p}_j) - I_l(\boldsymbol{p}s_l + \boldsymbol{W}\boldsymbol{p}_j) - m, \tag{10.17}$$

where the scalar $s_l = 0.5^l$ accounts for the down-sampling between the images of the image pyramid. Furthermore, by subtracting the mean intensity error m we can account for inter-frame illumination changes.

For regular patch alignment, the squared error terms of eq. (10.17) can be summed over all image levels and patch pixels and combined into a single Gauss-Newton optimization in order to find the optimal patch coordinates. However, the direct use of such a large number of error terms within an EKF would make it computationally intractable. In order to tackle this issue we apply a QR-decomposition on the linear equation system resulting from stacking all error terms in eq. (10.17) together for given estimated coordinates $\hat{\boldsymbol{p}}$:

$$\bar{\boldsymbol{b}}(\hat{\boldsymbol{p}}) = \bar{\boldsymbol{A}}(\hat{\boldsymbol{p}})\delta\boldsymbol{p}, \tag{10.18}$$

where $\bar{\boldsymbol{A}}(\hat{\boldsymbol{p}})$ can be computed based on the patch intensity gradients. Independent of the rank of the matrix $\bar{\boldsymbol{A}}(\hat{\boldsymbol{p}})$, the QR-decomposition of $\bar{\boldsymbol{A}}(\hat{\boldsymbol{p}})$ can be used to obtain an *equivalent* reduced linear equation system:

$$\boldsymbol{b}(\hat{\boldsymbol{p}}) = \boldsymbol{A}(\hat{\boldsymbol{p}})\delta\boldsymbol{p}, \tag{10.19}$$

with $\boldsymbol{A}(\hat{\boldsymbol{p}}) \in \mathbb{R}^{2\times 2}$ and $\boldsymbol{b}(\hat{\boldsymbol{p}}) \in \mathbb{R}^2$. Since we assume that the additive noise magnitude on the intensities is equal for every patch pixel we can leave it out of the above derivations (it will remain constant for every entry).

One interesting remark is, that due to the scaling factor s_l in eq. (10.17), error terms for higher image levels will a have weaker corrective influence on the filter state or the patch alignment. On the other hand, their increased robustness w.r.t. image blur or bad initial alignment strongly increases the robustness of the overall alignment method for multilevel patch features.

3.3 Feature Detection and Removal

The detection of new features is based on a standard fast corner detector which provides a large amount of candidate feature locations. After removing candidates which are close to current tracked features, we compute an adapted Shi-Tomasi score for selecting new features which will be added to the state. The adapted Shi-Tomasi score basically considers the combined Hessian on multiple image levels, instead of only a single level. It directly approximates the Hessian of the above gradient matrix with $\boldsymbol{H} = \bar{\boldsymbol{A}}^T(\hat{\boldsymbol{p}})\bar{\boldsymbol{A}}(\hat{\boldsymbol{p}})$ and extracts the minimal eigenvalue. The advantage is that a high score is directly correlated with the alignment accuracy of the corresponding multilevel patch feature. Instead of returning the minimal eigenvalue, the method can return other eigenvalue based scores like the 1- or 2-norm. This could be useful in environments with scarce corner data, whereby the presented filter could be complemented by available edge-shaped features. Finally, the detection process is also coupled to a bucketing technique in order to achieve a good distribution of the features within the image frame.

Table 10.1: Timings of Presented Approach per Processed Image

Tot. Features	10	20	30	40	50
Timing [ms]	6.65	10.50	14.87	21.48	29.72

Due to the fact that we can only track a limited number of features in the EKF, we have to implement a landmark management system to ensure that only reliable landmarks are inserted and kept in the filter state. Here, we fall back to heuristic methods, where we compute quality scores in order to decide whether a feature should be kept or not. The overall idea is to evaluate a local (only last few frames) and a global (how good was the feature tracked since it has been detected) quality score and remove the features below a certain threshold. Using an adaptive threshold we can control the total amount of features which are currently in the frame.

4 Results and Discussion

4.1 Experimental Setup

The data for the experiments were recorded with the VI-Sensor [104], equipped with two time-synchronized, global-shutter, wide-VGA 1/3 inch imagers in a fronto-parallel stereo configuration. The cameras are equipped with lenses with a diagonal field of view of 120 degrees and are factory-calibrated by the manufacturer for a standard pinhole projection model and a radial-tangential distortion model. The imagers are hardware time-synchronized to the IMU to ensure mid-exposure IMU triggering. In the context of this work only the image stream from one camera is required.

Ground truth is provided through an external motion capture system for the pose of the sensor. The rate of the IMU measurements is 200 Hz and the image frame rate is 20Hz. The employed IMU is an industrial-grade ADIS 16448, with an angular random walk of $0.66\,\mathrm{deg}/\sqrt{\mathrm{Hz}}$ and a velocity random walk of $0.11\,\mathrm{m/s}/\sqrt{\mathrm{Hz}}$. The maximal number of features in the state is set to 50 and the algorithm is run using image pyramids with 4 levels. Whenever possible, covariance parameters are selected based on hardware specifications. Strong tuning was not necessary, and the framework works well for a large range of parameters. The initial IMU-camera extrinsics are only roughly guessed (the translation is set to zero), and the initial inverse distance parameter for a feature is set to $0.5\,\mathrm{m}^{-1}$ with a standard deviation of $1\,\mathrm{m}^{-1}$. A screenshot of the running framework is depicted in fig. 10.2.

4.2 Experiment with Slow Motions

An experiment with slow to medium fast hand-held motions of about 1 min was carried out to evaluate the performance of the framework with different numbers of total features (from 10 to 50 in steps of 10). The performance was assessed by computing the relative position error w.r.t. the traveled distance [50]. Furthermore we compared the obtained results to a batch optimization framework along the lines of [83]. Figure 10.3 depicts the extracted relative error values. The achieved performance tends to be similar to the one of the batch optimization framework and often achieves slightly

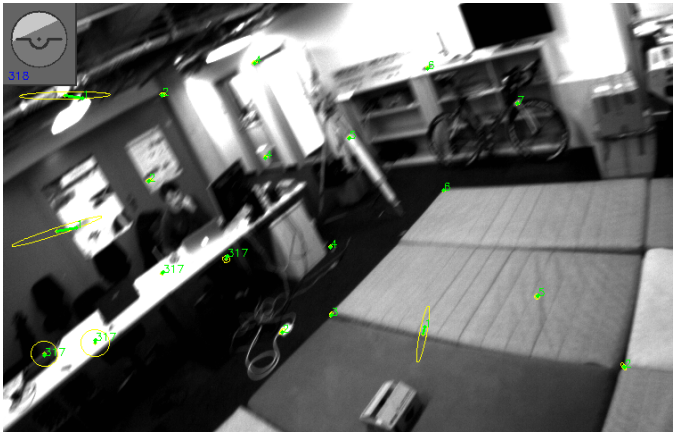

Figure 10.2: Screenshot of the running visual-inertial odometry framework. The 2-σ uncertainty ellipses of the predicted feature locations are in yellow, whereby only features which are newly initialized (stretched ellipses) and features which re-enter the frame have a significant uncertainty. Green points are the locations after the update step. Green numbers are the tracking counts (1 for newly initialized features). In the top left a virtual horizon is depicted.

higher accuracy. While these results depend on the specific dataset and parameter tuning, we also have to mention that the relatively high rotational motion (average of around 1.5 rad/s) favors approaches which can handle arbitrarily short feature tracks. Given the *undelayed* initialization of feature within our approach, the resulting filter is able to extract visual information from a feature's second observation onwards.

Surprisingly, the performance was relatively independent of the total amount of tracked features. A significant drop in accuracy could only be observed with feature counts below 20. This observation can have different reasons. One could be the type of sensor motions with relatively high rotational rates, which can lead to more bad features or outliers. Another point is also that our approach considers $256 = 4 \times 8 \times 8$ intensity errors per tracked features and thus we cannot directly compare to standard feature tracking based visual odometry frameworks, which typically require much higher feature counts. More in-depth evaluation of this effect will be part of future work. The timings of the proposed framework are listed in table 10.1 for a single core of an Intel i7-2760QM. The setup with 50 features uses an average processing time of 29.72 ms per processed image and can thus easily be run at 20 Hz.

4.3 Experiment with Fast Motions

Here, we evaluate the robustness of the proposed approach w.r.t. very fast motions. We recorded a hand-held dataset with mean rotational rate of around 3.5 rad/s and

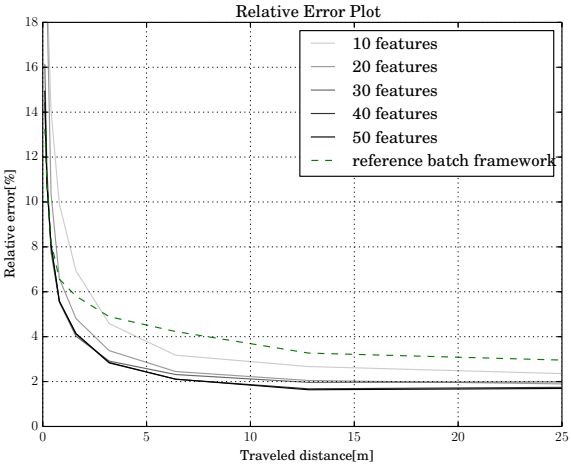

Figure 10.3: Gray lines are the relative errors of the presented approach, where the darkest lines corresponds to 50 features and the brightest line to 10 features respectively. The dashed green line represents the performance of the reference batch optimization framework.

with peaks of up to 8 rad/s. The motion capture system exhibited a relative high number of bad tracking, whereby we filtered them out as good as possible. We investigate the tracking performance of the attitude and of the robocentric velocities, where the corresponding estimates with 3σ-bounds are plotted in figs. 10.4 and 10.5 respectively. It can clearly be seen that the estimates nicely fit the ground truth data from the motion capture. As known from previous work the inclination angles and the robocentric velocities of visual-inertial setups are fully observable [85], and we can nicely observe the initial decrease of the corresponding covariance (especially when the system gets excited). On the other hand the yaw angle is unobservable and drifts slowly with time.

Figures 10.6 and 10.7 depict the estimation of the calibration parameters. Again, the estimates together with their 3σ-bounds are plotted. Depending on the excitation of the system the estimated values converge relatively quickly. It can be observed, that the translational term of the IMU-camera calibration requires a lot of rotational motion in order to converge appropriately. For the presented experiment, the accelerometer bias exhibits the worse convergence rate but is still within a reasonable range.

Furthermore, we also observed a divergence mode for the presented approach. It can occur when the velocity estimate diverge, e.g., due to missing motion or too many outliers. The problem is then, that the filter attempts to minimize the effect of the

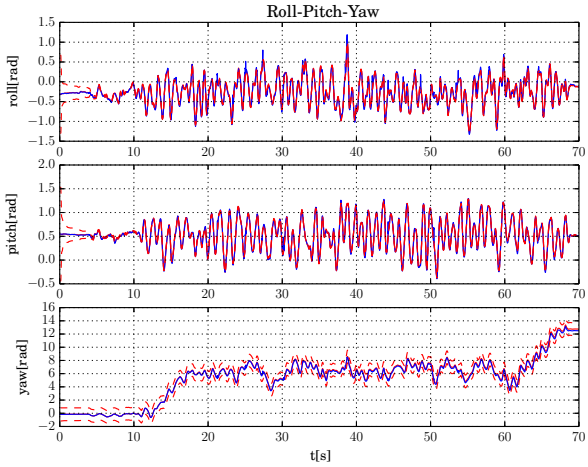

Figure 10.4: Euler angle estimates. Red: estimate, blue: motion capture, red dashed: 3σ-bound. Only the yaw angle is not observable and exhibits a growing covariance. The inclination angles (roll and pitch) exhibit a high quality tracking accuracy.

erroneous velocity on the bearing vectors by setting the distance of the features to infinity. This again lowers any corrective effect on the diverging velocity resulting in further divergence. All in all this was very rarely observed for regular usage, especially if the system was properly excited at the start.

4.4 Flying Experiments

Implementing the framework on-board a UAV with a forward oriented visual-inertial sensor, we also performed preliminary experiments on a real robot. The special aspect here is that the visual-inertial odometry framework was initialized on the ground without any previous calibration motions, i.e. the calibration parameters had to converge during take-off. The output of the filter was directly used for feedback control of the UAV. Figure 10.8 depicts the estimated position output of the framework during take-off, flying and landing. If compared to the motion capture system the filter exhibits a certain offset which can be mainly attributed to the online calibration of the filter.

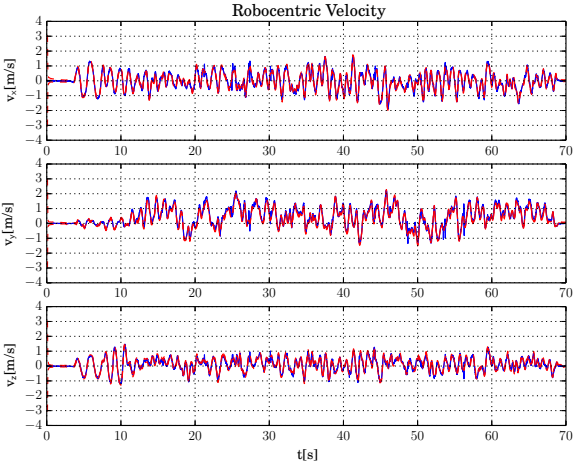

Figure 10.5: Velocity estimates. Red: estimate, blue: motion capture, red dashed: 3σ-bound. The robocentric velocity is fully observable and thus exhibits a bounded uncertainty. It very nicely tracks the reference from the motion capture system (and probably also exhibits a higher precision).

5 Conclusion

In this paper we presented a visual-inertial filtering framework which uses direct intensity errors as visual measurements within the extended Kalman filter update. By choosing a fully robocentric representation of the filter state together with a numerically minimal bearing/distance representation of features, we avoid major consistency problems while exhibiting accurate tracking performance and high robustness. Especially in difficult situations with very fast motions or outliers the presented approach manages to keep track of the state with only minor drift of the yaw and position estimates. The framework can be run on-board a UAV with a feature count of 50 at a framerate of 20 Hz and was used to stabilize the flight of a UAV from take-off to landing.

Future work will include more extensive evaluation of the multilevel patch features in context of intensity error based visual-inertial odometry frameworks. Furthermore we would also like to try to extend the online calibration in order to include the camera intrinsics. Also, the framework could be relatively easily adapted in order to handle multiple cameras. This could improve the filter performance, especially for cases with lack of translational motion. Another option to avoid divergence would be to use some heuristics based methods in order to detect such modes and to add zero-velocity pseudo-measurements in order to stabilize the filter. A detailed observability analysis

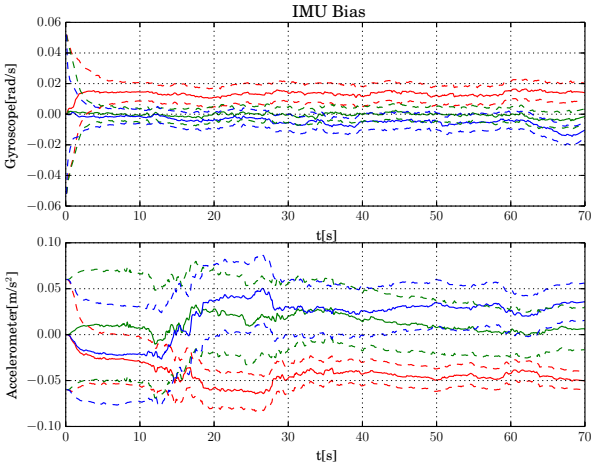

Figure 10.6: Estimated IMU biases. Top: gyroscope bias (red: x, blue: y, green: z), bottom: accelerometer bias (red: x, blue: y, green: z). The gyroscope biases exhibit a better convergence than the accelerometer biases, probably due to the more direct link of rotational rates to visual errors.

could also be performed, where the dependency of unobservable modes w.r.t. sensor motions would be of high interest.

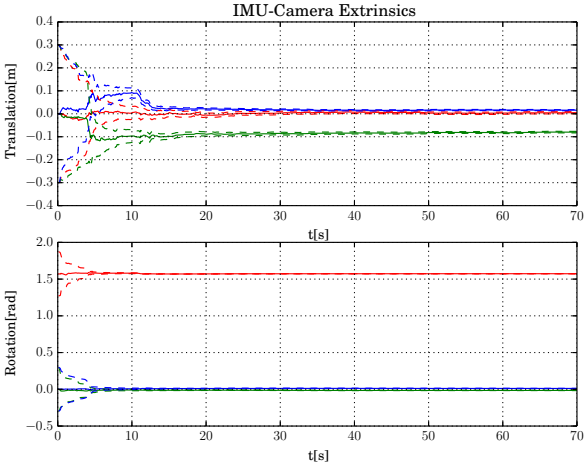

Figure 10.7: Estimated IMU-camera extrinsics. Top: translation (red: x, blue: y, green: z), bottom: orientation (red: yaw, blue: pitch, green: roll). Especially when sufficiently excited, the estimates converge quickly. The reached values correspond approximately to the ones obtained from an offline calibration.

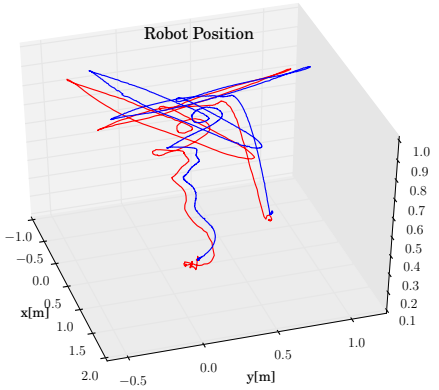

Figure 10.8: Estimated trajectory (red) on-board a UAV compared to groundtruth (blue) from the motion capture system. During take-off, flying, and landing the output of the filter is used to stabilize and control the UAV. Calibration is performed online.

Paper VII

IEKF-based Visual-Inertial Odometry using Direct Photometric Feedback

Michael Bloesch, Michael Burri, Sammy Omari, Marco Hutter, Roland Siegwart

Abstract

This paper presents a visual-inertial odometry framework which tightly fuses inertial measurements with visual data from one or more cameras, by means of an iterated extended Kalman filter (IEKF). By employing image patches as landmark descriptors, a photometric error is derived, which is directly integrated as an innovation term in the filter update step. Consequently, the data association is an inherent part of the estimation process and no additional feature extraction or matching processes are required. Furthermore, it enables the tracking of non-corner shaped features, such as lines, and thereby increases the set of possible landmarks. The filter state is formulated in a fully robocentric fashion, which reduces errors related to nonlinearities. This also includes partitioning of a landmark's location estimate into a bearing vector and distance and thereby allows an undelayed initialization of landmarks. Overall, this results in a compact approach which exhibits a high level of robustness with respect to low scene texture and motion blur. Furthermore, there is no time-consuming initialization required and pose estimates are available starting at the second image frame. We test the filter on different real datasets and compare it to other state-of-the-art visual-inertial frameworks. The experimental results show that robust localization with high accuracy can be achieved with this filter-based framework.

Published in:
International Journal of Robotics Research, SAGE 2017

1 Introduction

Robust and high-bandwidth estimation of ego-motion is a key factor to enable the operation of autonomous robots. For dynamically controlled robots, such as aerial vehicles or legged robots, a reliable state estimate is essential: Failures of the state estimator can quickly lead to damage of the hardware and its surroundings. Thus, as autonomous robots become more capable and extend their range of applications, it is essential that the corresponding ego-motion estimation can perform well in increasingly difficult environments. The corresponding selection of sensors should be kept as lightweight and low-cost as possible in order to employ them on a wide range of robotic systems robotic systems. Furthermore, in the context of vision-based estimation, extreme conditions such as strongly varying lighting, missing texture, fast motion, or dynamic objects may need to be accounted for.

Past research has shown that combining the complementary information from an Inertial Measurement Unit (IMU) and visual sensors can be a very capable approach in terms of accuracy and reliability. Consequently this approach has been successfully applied to robotic systems such as unmanned aerial robots ([119, 139]) or legged robots ([92, 128]). Since assessing the precision of an algorithm is often simpler than evaluating its robustness, many researchers have focused on optimizing the accuracy of their approaches. The evaluation is typically done by measuring the accumulated position error over given traveled distances. Depending on the experimental setup, state-of-the-art algorithms reduce position errors to 0.1% of the traveled distance ([41, 84, 136]). Such a demonstration of high accuracy can serve as surrogate for the well-functioning of an approach. However, all odometry frameworks inherently suffer from drift and, if the primary goal is localization accuracy, a back-end framework doing global mapping, re-localization and loop closure will be indispensable (e.g. [90]). Furthermore, if the ego-motion estimation is employed within a feedback loop on an autonomous robot, other aspects like reliability and estimation time-delay become important as well.

The well-established Kalman Filtering techniques represent sensor fusion frameworks that allow computationally efficient and high-bandwidth state estimation. Due to the inherent marginalization, the filter states at each timestep can refer to different physical quantities, e.g., a landmark's position can be estimated w.r.t. the moving sensor frame (and thereby represent a varying quantity over time). This enables the use of a fully robocentric formulation of the state and thereby reduces observability/nonlinearity related issues ([21]). To mitigate the problem of intrinsic unobservability of a landmark's initial distance from the observer, the landmark position can be parameterized by its bearing vector and distance ([98]). Consequently a landmark's distance can be initialized with a high uncertainty without affecting its bearing vector estimate (which can be initialized with a low uncertainty). Especially for scenarios with fast motions and short feature tracks, this becomes invaluable as it allows a seamless initialization of landmarks and thereby the extraction of visual information out of a landmark's second observation onwards. Bearing vectors can be represented as members of a 2D-manifold, with a corresponding Lie-Algebra being used for filtering, leading to a minimal and consistent representation of bearing vectors and their uncertainty ([59]).

The proposed approach combines an iterated extended Kalman filter (IEKF), a fully

robocentric formulation of visual-inertial odometry, and a photometric error model. This is achieved by associating every landmark with a multilevel patch feature, where the innovation term is derived by projecting the patch into the current image and computing the photometric error for every patch pixel. To keep the computational effort tractable, a QR-decomposition based reduction is applied for obtaining an *equivalent* 2D innovation term per observed landmark. This method takes into account the local texture of a landmark and thereby gains more information along the directions where the patch gradients are stronger. In addition, this offers the possibility to track non-corner shaped features, such as lines, increasing the set of possible image features which is beneficial in scenarios with missing texture.

In contrast to our previous work ([14]), which implemented a regular extended Kalman filter (EKF), the employed IEKF allows per-landmark iterative updates. This inherently takes care of landmark tracking where a landmark's position estimate is iteratively updated by simultaneously considering the current IMU-based prior and the observed patch texture. To the best of our knowledge, this *tight* combination of data association and information fusion is novel for visual-inertial odometry. This approach is similar to dense visual algorithms which avoid separate data association through image alignment techniques. All in all, this yields a *fully robocentric* and *direct* visual-inertial odometry framework which runs in real-time on computationally constrained platforms. To increase robustness and usability, we implement multi-camera support (with or without overlapping field of view) and enable online calibration of camera-IMU extrinsics. An in-depth derivation and evaluation of the framework is provided, including experiments on publicly available datasets [20]. Our framework, which we refer to as Rovio (RObust Visual-Inertial Odometry), is implemented in C++ and is available as open-source software [1].

2 Related Work

Within the field of computer vision, [28] proposed one of the first real-time 3D monocular localization and mapping frameworks. Similarly to the work in this paper, the author made use of an EKF framework where he co-estimates the absolute position of 3D landmarks. Since then, various research groups have contributed improvements and proposed further approaches. A key issue is to improve the consistency of the estimation framework that is affected by its inherent nonlinearity ([21, 75]). One approach is to make use of a robocentric representation for the tracked landmarks and thereby significantly reduce the effect of nonlinearities ([21, 25]). As an alternative, [61] propose the use of a so-called observability constrained extended Kalman filter, whereby the inconsistencies can be avoided by using special linearization points while evaluating the system Jacobians.

A somewhat related problem is the choice of the specific representation of a landmark's location. Since the depth of a newly detected landmark is unknown for monocular setups, the initial 3D location estimate exhibits a high (infinite) uncertainty along the corresponding axis. To integrate this landmark from the beginning into the estimation framework, [98] proposed the use of an inverse-depth parametrization (IDP). They parametrize each landmark location by the camera position where the landmark

[1] https://github.com/ethz-asl/rovio

was initially detected, by a bearing vector (parametrized with azimuth and elevation angles), as well as the inverse depth of the landmark. The increase in consistency for the IDP and other parametrization methods was further analyzed and confirmed by [126].

While most standard visual odometry approaches are based on detected and tracked point landmarks as source of visual information, so-called *direct* approaches directly use the image intensities in their estimation framework. [71] propose to model the environment as a collection of planar patches and to derive a corresponding photometric error between camera frames. Their work is similar to ours in that they also embed the photometric error directly into a filtering framework (but they do not use any inertial data which limits them to slow motions). [97] also track locally planar image patches in a filter-based SLAM framework. By employing gradient-based image alignment, they also co-estimate surface normals but keep data association separated from the subsequent EKF-based information fusion. [122] also use planar regions and minimize the photometric error with respect to a reference frame in order to estimate the relative motion as well as other parameters like illumination parameters and patch normals. They then subsequently merge the output in an EKF. By employing highly optimized SIMD (Single Instruction Multiple Data) implementations, first real-time, CPU-based approaches for semi-dense motion estimation using a monocular camera ([33, 40]) have recently been proposed.

Incorporating inertial measurements in the estimation can significantly improve the robustness of the system, provides the estimation process with the notion of gravity, and allows for a more accurate and high bandwidth estimation of the velocities and rotational rates. By adapting the original EKF proposed by [28], additional IMU measurements can be relatively simply integrated into the ego-motion estimation, whereby calibration parameters can be co-estimated online ([73, 79]). [84] describe a *tightly* coupled approach in which the robot trajectory and sparse 3D landmarks are estimated in a joint optimization problem using inertial error terms as well as the reprojection error of the tracked landmarks in the camera images. This is done in a windowed bundle adjustment approach over a set of keyframe images and a temporal inertial measurement window. Similarly, [99] estimate the trajectory in an IMU-driven filtering framework using the reprojection error of 3D landmarks as measurement updates. Instead of adding the landmarks to the filter state, they immediately marginalize them out using a nullspace decomposition, thus leading to a small filter state size. Since inertial measurements are often obtained at a higher rate than image data, methods for combining multiple inertial measurements are desirable to reduce the computational costs. [41] have presented a concise IMU measurements pre-integration method such that they can be efficiently included in a factor graph framework. Recently, [136] have extended their previous work on dense visual odometry ([33]) in order to integrate inertial measurements. They minimize a joint energy term composed of visual and inertial error terms in order to estimate the ego-motion of their sensor.

Probably the most comparable work to ours was developed by [132], who implemented an EKF-based framework for merging patch-based photometric errors with IMU measurements. They parameterize their landmarks by the pose of the camera when the landmark was detected as well as the corresponding bearing vector and inverse depth (analogously to [98]). Our work differs in that it uses a fully robocentric

formulation of the current state, which has various implications on the filtering and visual processing framework. We also integrate a QR-decomposition based measurement space reduction and perform per-landmark update iterations, which are both key to the efficiency and accuracy of our system.

3 Prerequisites on Rotations and Unit Vectors

3.1 Notation

For better readability and comprehensibility, we give a brief overview of the employed notations and the algebra of 3D rotations and unit vectors. Three different coordinate frames are used throughout the paper: the inertial world coordinate frame, $\mathcal{I}$, the IMU fixed coordinate frame, $\mathcal{B}$, as well as the camera fixed coordinate frame, $\mathcal{C}$. Only in section 6, where multi-camera setups are discussed, the distinction between the different camera frames will be made. The origin associated with a specific coordinate frame is denoted by the same symbol. In this context, a term of the form ${}_{\mathcal{I}}\boldsymbol{r}_{\mathcal{BC}}$ denotes the coordinates of a vector from the origin of $\mathcal{B}$ to the origin of $\mathcal{C}$, expressed in the coordinate frame $\mathcal{I}$. Furthermore, $\boldsymbol{q}_{\mathcal{BI}}$ is employed in an abstract manner for representing the rotation between a frame $\mathcal{I}$ and $\mathcal{B}$ (the actual implementation is mainly based on unit quaternions). A good way to think of a rotation is as a mapping $\boldsymbol{q}_{\mathcal{BI}} : \mathbb{R}^3 \to \mathbb{R}^3$ between the two associated coordinate frames: Given a physical vector $\boldsymbol{r}_{\mathcal{BC}}$, a rotation maps the corresponding coordinates from the right index frame to the left index frame, e.g., ${}_{\mathcal{B}}\boldsymbol{r}_{\mathcal{BC}} = \boldsymbol{q}_{\mathcal{BI}}\,({}_{\mathcal{I}}\boldsymbol{r}_{\mathcal{BC}})$. We also employ the mapping $\boldsymbol{C}(\boldsymbol{q}) : SO(3) \to \mathbb{R}^{3\times 3}$ which is defined such that $\boldsymbol{q}(\boldsymbol{r}) \triangleq \boldsymbol{C}(\boldsymbol{q})\boldsymbol{r}$ and basically returns the 3×3 rotation matrix.

As further abbreviations, we use $\boldsymbol{v}_{\mathcal{B}}$ for denoting the absolute velocity of $\mathcal{B}$, and $\boldsymbol{\omega}_{\mathcal{IB}}$ for the vector describing the relative rotational velocity of the coordinate frame $\mathcal{B}$ w.r.t. the coordinate frame $\mathcal{I}$. In some cases we use further denotations like tildes (measurements) or hats (estimates) if we want to highlight a specific aspect of a quantity. The superscript $^\times$ is used to denote the skew symmetric matrix $\boldsymbol{v}^\times \in \mathbb{R}^{3\times 3}$ of a vector $\boldsymbol{v} \in \mathbb{R}^3$.

3.2 Representation of 3D Rotations

Since rotations are part of the special orthogonal group $SO(3)$ (with group operation $\otimes$), there is no direct notion of addition or subtraction (and consequently no differentiation either). Fortunately, since $SO(3)$ is a Lie group, a logarithmic and an exponential map exist which map to and from a corresponding Lie algebra $\mathbb{R}^3$:

$$\begin{aligned} \log :& SO(3) \to \mathbb{R}^3, \\ & \boldsymbol{q}_{\mathcal{BI}} \mapsto \log(\boldsymbol{q}_{\mathcal{BI}}) = \boldsymbol{\theta}_{\mathcal{BI}}, \end{aligned} \tag{11.1}$$

$$\begin{aligned} \exp :& \mathbb{R}^3 \to SO(3), \\ & \boldsymbol{\theta}_{\mathcal{BI}} \mapsto \exp(\boldsymbol{\theta}_{\mathcal{BI}}) = \boldsymbol{q}_{\mathcal{BI}}. \end{aligned} \tag{11.2}$$

There is a certain amount of freedom in selecting these maps. Here, we select the exponential and logarithmic maps such that $\boldsymbol{\theta}_{\mathcal{BI}}$ in the above equations coincides

with the passive rotation vector of the rotation $\boldsymbol{q}_{\mathcal{BI}}$. We can write the following identities (the last identity is known as Rodrigues' formula):

$$\exp(-\boldsymbol{\theta}) = \exp(\boldsymbol{\theta})^{-1}, \tag{11.3}$$

$$\exp(\boldsymbol{q}(\boldsymbol{\theta})) = \boldsymbol{q} \otimes \exp(\boldsymbol{\theta}) \otimes \boldsymbol{q}^{-1}, \tag{11.4}$$

$$\boldsymbol{C}(\boldsymbol{\theta}) = \boldsymbol{I} - \frac{\sin(\|\boldsymbol{\theta}\|)\boldsymbol{\theta}^{\times}}{\|\boldsymbol{\theta}\|} + \frac{(1-\cos(\|\boldsymbol{\theta}\|))\boldsymbol{\theta}^{\times^2}}{\|\boldsymbol{\theta}\|^2}. \tag{11.5}$$

The exponential and logarithmic maps can now be used to introduce a boxplus ($\boxplus$) and a boxminus ($\boxminus$) operator, which adopt the role of addition and subtraction operators for rotations ([59]). Using a slightly different notation, we define:

$$\begin{aligned} \boxplus : & SO(3) \times \mathbb{R}^3 \to SO(3), \\ & \boldsymbol{q}, \boldsymbol{\theta} \mapsto \exp(\boldsymbol{\theta}) \otimes \boldsymbol{q}, \end{aligned} \tag{11.6}$$

$$\begin{aligned} \boxminus : & SO(3) \times SO(3) \to \mathbb{R}^3, \\ & \boldsymbol{q}, \boldsymbol{p} \mapsto \log(\boldsymbol{q} \otimes \boldsymbol{p}^{-1}). \end{aligned} \tag{11.7}$$

Similarly to regular addition and subtraction, both operators fulfill the following identities (axioms proposed by [59]):

$$\boldsymbol{q} \boxplus \boldsymbol{0} = \boldsymbol{q}, \tag{11.8}$$

$$(\boldsymbol{q} \boxplus \boldsymbol{\theta}) \boxminus \boldsymbol{q} = \boldsymbol{\theta}, \tag{11.9}$$

$$\boldsymbol{q} \boxplus (\boldsymbol{p} \boxminus \boldsymbol{q}) = \boldsymbol{p}. \tag{11.10}$$

This approach distinguishes between actual rotations which are on $SO(3)$ (Lie group) and differences of rotations which lie on $\mathbb{R}^3$ (Lie algebra). The above operators take care of appropriately transforming the elements into their respective spaces and allow a smooth embedding of rotational quantities in filtering and optimization frameworks.

The definition of differentials involving rotation can be adapted by replacing the regular plus and minus operators by the above boxplus and boxminus operators. For instance the differential of a mapping $\boldsymbol{q}(x) : \mathbb{R} \to SO(3)$ can be defined as:

$$\frac{\partial}{\partial x}\boldsymbol{q}(x) := \lim_{\epsilon \to 0} \frac{\boldsymbol{q}(x+\epsilon) \boxminus \boldsymbol{q}(x)}{\epsilon}. \tag{11.11}$$

The same can be done for the other way round where we have a mapping $x(\boldsymbol{q}) : SO(3) \to \mathbb{R}$:

$$\frac{\partial}{\partial \boldsymbol{q}} x(\boldsymbol{q}) := \lim_{\epsilon \to 0} \begin{bmatrix} \frac{x(\boldsymbol{q} \boxplus (\boldsymbol{e}_1 \epsilon)) - x(\boldsymbol{q})}{\epsilon} \\ \frac{x(\boldsymbol{q} \boxplus (\boldsymbol{e}_2 \epsilon)) - x(\boldsymbol{q})}{\epsilon} \\ \frac{x(\boldsymbol{q} \boxplus (\boldsymbol{e}_3 \epsilon)) - x(\boldsymbol{q})}{\epsilon} \end{bmatrix}^T \tag{11.12}$$

where $\boldsymbol{e}_{1/2/3}$ are orthonormal basis vectors. This results in the following frequently-used derivatives (these may vary depending on conventions):

$$\partial/\partial t\left(\boldsymbol{q}_{\mathcal{BI}}(t)\right) = {}_{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{IB}}(t), \tag{11.13}$$

$$\partial/\partial \boldsymbol{q}\left(\boldsymbol{q}(\boldsymbol{r})\right) = \left(\boldsymbol{q}(\boldsymbol{r})\right)^{\times}, \tag{11.14}$$

$$\partial/\partial \boldsymbol{q}\left(\boldsymbol{q}^{-1}\right) = -\boldsymbol{C}(\boldsymbol{q})^{T}, \tag{11.15}$$

$$\partial/\partial \boldsymbol{q}\left(\boldsymbol{q} \otimes \boldsymbol{p}\right) = \boldsymbol{I}, \tag{11.16}$$

$$\partial/\partial \boldsymbol{q}\left(\boldsymbol{p} \otimes \boldsymbol{q}\right) = \boldsymbol{C}(\boldsymbol{p}), \tag{11.17}$$

$$\partial/\partial \boldsymbol{\theta}\left(\exp(\boldsymbol{\theta})\right) = \boldsymbol{\Gamma}(\boldsymbol{\theta}), \tag{11.18}$$

$$\partial/\partial \boldsymbol{q}\left(\log(\boldsymbol{q})\right) = \boldsymbol{\Gamma}^{-1}(\log(\boldsymbol{q})). \tag{11.19}$$

The derivative of the exponential map is given by the Jacobian $\boldsymbol{\Gamma}(\boldsymbol{\theta}) \in \mathbb{R}^{3\times3}$ which has the following analytical expression:

$$\boldsymbol{\Gamma}(\boldsymbol{\theta}) = \boldsymbol{I} - \frac{(1-\cos(\|\boldsymbol{\theta}\|))\boldsymbol{\theta}^{\times}}{\|\boldsymbol{\theta}\|^2} + \frac{(\|\boldsymbol{\theta}\| - \sin(\|\boldsymbol{\theta}\|))\boldsymbol{\theta}^{\times^2}}{\|\boldsymbol{\theta}\|^3}. \tag{11.20}$$

A more detailed discussion and derivations can be found in [16].

3.3 Representation of 3D Unit Vectors

While the above handling of rotations has been used similarly in previous filtering frameworks (e.g. [11, 85]), we extend the methodology to 3D unit vectors on the 2-sphere S^2. This is done analogously to [59], whereas we employ a parametrization yielding simple analytical derivatives and guarantee second order differentiability. A main issue with 3D unit vectors is to select orthonormal vectors for spanning the tangent space such that a suitable difference operator can be defined. Assigning orthonormal vectors to every point on the 2-sphere creates a vector field and as stated by the "hairy ball theorem", there is no continuous way of doing so over the full 2-sphere. To solve this issue we employ a rotation, $\boldsymbol{\mu} \in SO(3)$, as underlying representation for unit vectors and define the following quantities:

$$\boldsymbol{n}(\boldsymbol{\mu}) := \boldsymbol{\mu}(\boldsymbol{e}_z) \in S^2 \subset \mathbb{R}^3, \tag{11.21}$$

$$\boldsymbol{N}(\boldsymbol{\mu}) := [\boldsymbol{\mu}(\boldsymbol{e}_x), \boldsymbol{\mu}(\boldsymbol{e}_y)] \in \mathbb{R}^{3\times2}, \tag{11.22}$$

where $\boldsymbol{e}_{x/y/z} \in \mathbb{R}^3$ are the basis vectors of an arbitrary orthonormal coordinate system. The actual unit vector is given by $\boldsymbol{n}(\boldsymbol{\mu})$ which results when rotating $\boldsymbol{e}_z$ by $\boldsymbol{\mu}$ (if the context is clear we directly refer to the unit vector using $\boldsymbol{\mu}$). The matrix $\boldsymbol{N}(\boldsymbol{\mu})$ is composed of the rotated $\boldsymbol{e}_x$ and $\boldsymbol{e}_y$ and spans the tangent space. While such a construction of the tangent space is not deterministic since infinitely many rotations $\boldsymbol{\mu}$ provide the same unit vector $\boldsymbol{n}(\boldsymbol{\mu})$, we have the advantage that smooth transformations of the rotation $\boldsymbol{\mu}$ induce smooth transformations of the associated tangent space.

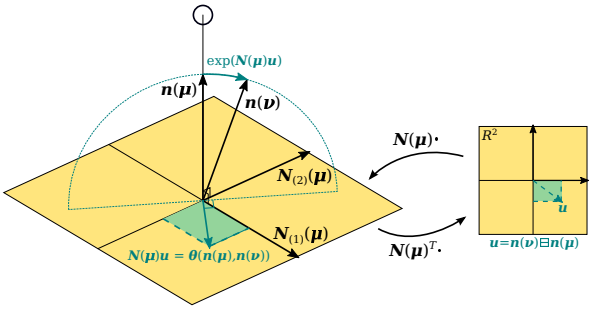

Figure 11.1: Representation of 3D unit vectors: The 3D unit vector $\boldsymbol{n}(\boldsymbol{\mu})$ is represented as the result of applying the rotation $\boldsymbol{\mu}$ onto the z-axis of an arbitrary inertial coordinate system. The images of the x- and y-axis are used to define an orthonormal plane to the unit vector. This plane then represents the tangent space used for the construction of the boxplus and boxminus operations. The boxminus operator takes two 3D unit vectors and represents their difference in $\mathbb{R}^2$. Conversely, the boxplus operator takes an element from $\mathbb{R}^2$ and applies it on a 3D unit vector.

The tangent space can be used to define the following boxplus and boxminus operators:

$$\begin{aligned} \boxplus : & SO(3) \times \mathbb{R}^2 \to SO(3), \\ & \boldsymbol{\mu}, \boldsymbol{u} \mapsto \exp(\boldsymbol{N}(\boldsymbol{\mu})\boldsymbol{u}) \otimes \boldsymbol{\mu}, \end{aligned} \tag{11.23}$$

$$\begin{aligned} \boxminus : & SO(3) \times SO(3) \to \mathbb{R}^2, \\ & \boldsymbol{\nu}, \boldsymbol{\mu} \mapsto \boldsymbol{N}(\boldsymbol{\mu})^T \boldsymbol{\theta}(\boldsymbol{\mu}, \boldsymbol{\nu}), \end{aligned} \tag{11.24}$$

where $\boldsymbol{\theta}$ maps two unit vectors to the minimal rotation vector between them:

$$\boldsymbol{\theta}(\boldsymbol{n}(\boldsymbol{\mu}), \boldsymbol{n}(\boldsymbol{\nu})) = \frac{\operatorname{acos}(\boldsymbol{n}(\boldsymbol{\nu})^T \boldsymbol{n}(\boldsymbol{\mu}))}{\|\boldsymbol{n}(\boldsymbol{\nu}) \times \boldsymbol{n}(\boldsymbol{\mu})\|} \boldsymbol{n}(\boldsymbol{\nu}) \times \boldsymbol{n}(\boldsymbol{\mu}). \tag{11.25}$$

A visualization of the 2-sphere and the tangent space for a specific $\boldsymbol{\mu}$ is given in Figure 11.1.

The concept is slightly more complicated than in the case of 3D rotations since we truly over-parameterize a 3D unit vector (no constraint is imposed on the underlying rotation). To overcome this, we use a different notion of equivalence where we define that two unit vector parametrizations $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are equivalent ($\boldsymbol{\mu} \sim \boldsymbol{\nu}$) *iff* $\boldsymbol{n}(\boldsymbol{\mu}) = \boldsymbol{n}(\boldsymbol{\nu})$. With this, the axioms proposed by [59] are again fulfilled:

$$\boldsymbol{\mu} \boxplus \boldsymbol{0} = \boldsymbol{\mu}, \tag{11.26}$$

$$(\boldsymbol{\mu} \boxplus \boldsymbol{u}) \boxminus \boldsymbol{\mu} = \boldsymbol{u}, \tag{11.27}$$

$$\boldsymbol{\mu} \boxplus (\boldsymbol{\nu} \boxminus \boldsymbol{\mu}) \sim \boldsymbol{\nu}. \tag{11.28}$$

A technical detail with this parametrization is that whenever representing a difference, $\boldsymbol{u} \in \mathbb{R}^2$, we have to keep track of the corresponding tangent space. Mathematically, if we have two rotations $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ with $\boldsymbol{\mu} \sim \boldsymbol{\nu}$, it does not necessarily follow that $(\boldsymbol{\mu} \boxplus \boldsymbol{u}) \sim (\boldsymbol{\nu} \boxplus \boldsymbol{u})$.

Similarly to the derivatives given in section 3.2, the most commonly used derivatives for terms involving 3D unit vectors are given by:

$$\begin{aligned} \partial/\partial t\left(\boldsymbol{\mu}(t)\right) = & -\boldsymbol{N}(\boldsymbol{\mu}(t))^T \boldsymbol{n}(\boldsymbol{\mu}(t))^\times \\ & \cdot \partial/\partial t\left(\boldsymbol{n}(\boldsymbol{\mu}(t))\right), \end{aligned} \tag{11.29}$$

$$\partial/\partial \boldsymbol{\mu}\left(\boldsymbol{n}(\boldsymbol{\mu})\right) = \boldsymbol{n}(\boldsymbol{\mu})^\times \boldsymbol{N}(\boldsymbol{\mu}), \tag{11.30}$$

$$\partial/\partial \boldsymbol{\mu}\left(\boldsymbol{N}(\boldsymbol{\mu})^T \boldsymbol{r}\right) = -\boldsymbol{N}(\boldsymbol{\mu})^T \boldsymbol{r}^\times \boldsymbol{N}(\boldsymbol{\mu}). \tag{11.31}$$

The first identity relates the time derivative of a 3D unit vector on its manifold to its time derivative in the 3D vector space. The second expression is the derivative of the unit vector in 3D w.r.t. to its minimal 2D representation. Those identities can be very useful when computing Jacobians, whereby the chain rule can be applied for computing the derivatives of more complex terms. An example will be provided when discussing the process model of the bearing vector state of 3D landmarks (see section 5.3 and section 9).

All in all, the proposed unit vector parametrization yields analogous advantages as obtained when employing the well established minimal 3D rotation parametrization. This includes a singularity-free parametrization which comes with relatively simple differentials. Furthermore the parametrization of the tangent space is orthogonal and the direction of the boxminus operation is in accordance with the shortest path between two given unit vectors (taking a step along $\boldsymbol{\nu} \boxminus \boldsymbol{\mu}$ is optimal for going from $\boldsymbol{\mu}$ to $\boldsymbol{\nu}$, see Figure 11.1). Other parametrizations, such as azimuth and elevation angles, do not meet these properties and often exhibit singular configurations.

4 Multilevel Patches and Photometric Error

4.1 Multilevel Patch Features

Along the lines of other landmark-based visual odometry approaches ([28]) we model landmarks as distinguished stationary 3D locations in the environment. Each landmark is associated with a multilevel patch feature $P = \{P_0, \ldots, P_L\}$, which is composed of multiple $n \times n$ image patches, P_l, extracted at the projected landmark location on image level l. In the current default implementation we extract 6×6 image patches on the second and third pyramid level (down-sampling factor of 2). These parameters can and should be adapted to the actual hardware setup and application scenario. An example is given in Figure 11.2. The simultaneous use of multiple pyramid levels leads to cross-correlations between the pixel intensities. These are not explicitly modeled but can be handled to a certain extent by tuning the corresponding error weighting.

In comparison to a standard feature descriptor, a patch-based descriptor allows to compute a photometric error and thereby to avoid the use of reprojection errors. Taking the information of every pixel gives much richer information about the environment, which not only helps improving the robustness in bad lighting conditions, but

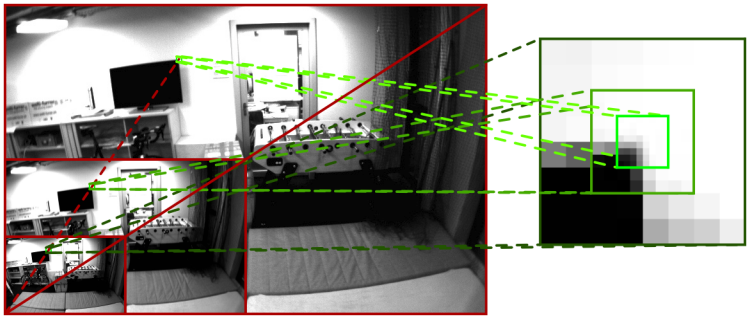

Figure 11.2: The construction of a multilevel patch out of an image pyramid. Here each single patch is composed of 8×8 pixels and 3 pyramid levels are depicted. These settings may vary in the actual implementation.

also inherently takes into account the texture of the tracked image patch. For instance, it enables the integration of edge-shaped features, whereby the gained information is along the perpendicular direction of the edge. In comparison, reprojection error based approaches typically attempt to minimize the distance between the predicted and detected feature location. This ignores the local texture around the landmark and, if no additional measures are taken, all landmarks are weighted equally.

4.2 Projection Model and Linear Warping

Given the bearing vector $\boldsymbol{\mu}$ of a landmark, the pixel coordinates in a specific camera frame can be retrieved by using the camera model $\boldsymbol{\pi}$. Assuming a known intrinsic calibration, the pixel coordinates $\boldsymbol{p}$ can directly be expressed by $\boldsymbol{p} = \boldsymbol{\pi}(\boldsymbol{\mu})$. If the camera is moving, the feature moves through the image and is seen from a different perspective. To account for a certain patch distortion effect, a linear warping matrix is tracked with each feature. This is done by concatenating all Jacobians when transforming a landmark location. For instance, if we detect a feature in some frame at pixel $\boldsymbol{p}_1$, transform the corresponding bearing vector $\boldsymbol{\mu}_1 = \boldsymbol{\pi}^{-1}(\boldsymbol{p}_1)$ with a process model $\boldsymbol{\mu}_2 = \boldsymbol{f}(\boldsymbol{\mu}_1)$, and then re-project the bearing vector in a subsequent frame $\boldsymbol{p}_2 = \boldsymbol{\pi}(\boldsymbol{\mu}_2)$, we obtain the following linear warping matrix:

$$\boldsymbol{D} = \frac{\partial \boldsymbol{\pi}(\boldsymbol{\mu}_2)}{\partial \boldsymbol{\mu}_2} \frac{\partial \boldsymbol{f}(\boldsymbol{\mu}_1)}{\partial \boldsymbol{\mu}_1} \frac{\partial \boldsymbol{\pi}^{-1}(\boldsymbol{p}_1)}{\partial \boldsymbol{p}_1} \in \mathbb{R}^{2\times 2}. \tag{11.32}$$

In essence, this maps the two patch axes from the point of patch extraction (which where aligned with the image axes) to the two distorted patch axes in the projection image. This approach tracks the distortion locally around the patch and ignores any larger scale information like the geometric shape of a patch. To avoid large distortions and the accumulation of errors, the patches a re-extracted regularly and the warping matrix is reset to identity.

4.3 Photometric Error and Patch Alignment

The photometric error between a given multilevel patch feature and a specific image is computed by extracting a warped patch at the estimated location and evaluating the pixel-wise intensity error. For a given multilevel patch feature (with coordinates $\boldsymbol{p}$ and multilevel patch $P = \{P_0, \dots, P_L\}$) at a specific image level l and patch pixel $\boldsymbol{p}_j$, the photometric error can be formalized as follows:

$$e_{l,j}(\boldsymbol{p}, P, I, \boldsymbol{D}) = P_l(\boldsymbol{p}_j) - a\ I_l(\boldsymbol{p}\, s_l + \boldsymbol{D}\boldsymbol{p}_j) - b, \tag{11.33}$$

where I_l is the image at the pyramid level l and $s_l = 0.5^l$ is a scaling factor to account for the down-sampling. The linear warping matrix $\boldsymbol{D}$ is used to map patch pixel coordinates to image coordinates. Furthermore, inter-frame illumination changes are taken into account by employing an affine intensity model composed of the scalars a and b (both get marginalized out). Figure 11.3 depicts the photometric error between a patch and its measurement in an image at a predicted location $\boldsymbol{p}$.

If we minimize the squared error terms for a multilevel patch, we obtain a patch alignment algorithm which is very similar to the well-known Kanade-Lucas-Tomasi (KLT) feature tracker ([89, 120]). A slight difference is given by the fact that we optimize over multiple image levels at once. The minimization can be solved by a Gauss-Newton method which iteratively linearizes the optimization problem around an estimated patch location $\hat{\boldsymbol{p}}$:

$$\boldsymbol{b}(\hat{\boldsymbol{p}} + \delta\boldsymbol{p}, P, I, \boldsymbol{D}) = \boldsymbol{A}(\hat{\boldsymbol{p}}, I, \boldsymbol{D})\delta\boldsymbol{p} + \boldsymbol{b}(\hat{\boldsymbol{p}}, P, I, \boldsymbol{D}), \tag{11.34}$$

where $\boldsymbol{b}(\hat{\boldsymbol{p}}, P, I, \boldsymbol{D})$ represents the stacked error terms from eq. (11.33) and $\boldsymbol{A}(\hat{\boldsymbol{p}}, I, \boldsymbol{D})$ the corresponding Jacobian. The corresponding normal equations are then given by:

$$\boldsymbol{A}(\hat{\boldsymbol{p}}, I, \boldsymbol{D})^T \boldsymbol{A}(\hat{\boldsymbol{p}}, I, \boldsymbol{D})\delta\boldsymbol{p} = -\boldsymbol{A}(\hat{\boldsymbol{p}}, I, \boldsymbol{D})^T \boldsymbol{b}(\hat{\boldsymbol{p}}, P, I, \boldsymbol{D}), \tag{11.35}$$

which can be solved for the correction $\delta\boldsymbol{p}$. This is analogous to one iteration step of the KLT feature tracker (but is not used as such in Rovio). In section 5.4 we will demonstrate how eq. (11.34) is leveraged into the innovation term of the employed IEKF.

Note that due to the scaling factor s_l in eq. (11.33), error terms for higher image levels will have a weaker corrective influence on the filter state or the patch alignment. On the other hand, they exhibit increased robustness w.r.t. image blur or bad initial alignment and thus strongly increase the robustness of the overall alignment method.

4.4 Detection and Scoring

The detection of new landmarks is based on the FAST corner detector ([115]) which provides a large amount of candidate feature locations. After removing candidates which are close to currently tracked features, we compute a patch gradient based score for selecting new features which are added to the state. This basically represents an adaptation of the Shi-Tomasi score ([120]) by considering the combined Hessian on

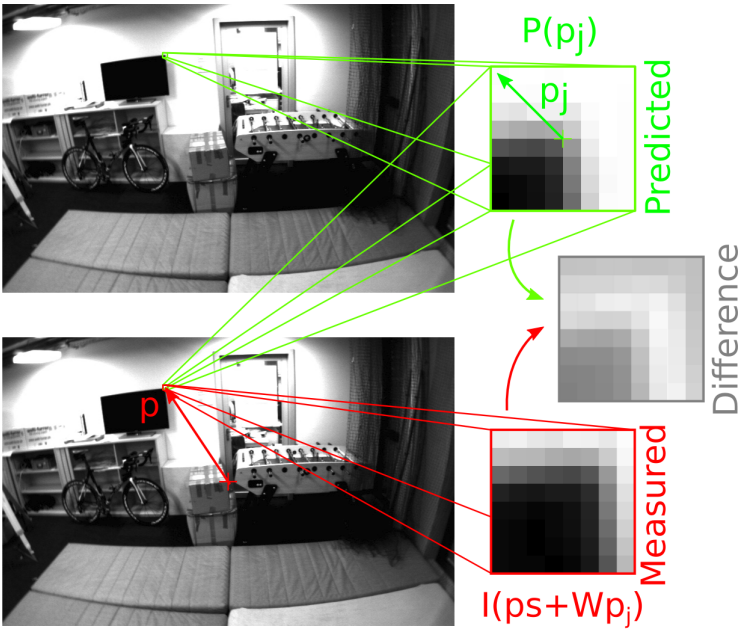

Figure 11.3: Illustration of the (signed) photometric error between a previously extracted patch (green) and its projection into an image (measured, red) at a predicted location $\boldsymbol{p}$. The bottom left grey tone of the difference patch represents 0. Only a single image level is depicted. This photometric error can directly be used as the innovation term in an IEKF.

multiple image levels, instead of only a single level. The combined Hessian can be directly retrieved from the normal equations (11.35):

$$\boldsymbol{H} = \boldsymbol{A}(\hat{\boldsymbol{p}}, I, \boldsymbol{D})^T \boldsymbol{A}(\hat{\boldsymbol{p}}, I, \boldsymbol{D}), \tag{11.36}$$

where the minimal eigenvalue of $\boldsymbol{H}$ corresponds to the adapted Shi-Tomasi score. The advantage is that a high score is directly correlated with the alignment accuracy of the corresponding multilevel patch feature. Instead of returning the minimal eigenvalue, the method can return other eigenvalue based scores like the 1- or 2-norm. This is useful in environments with scarce corner data, whereby also edge-shaped features can be considered. Finally, the detection process is also coupled with a bucketing technique to achieve a good distribution of the features within the camera frame.

5 Filter Framework

5.1 Iterated Extended Kalman Filtering

The regular Kalman filter can be interpreted as the recursive optimal solution to the maximum likelihood estimation problem formulated over two subsequent time steps ([5]). Analogously, the EKF can be associated with a nonlinear maximum likelihood estimation and can be shown to yield the same result as the first iteration step of a corresponding Gauss-Newton optimization. However, in contrast to its linear counterpart, the EKF cannot guarantee to retrieve the optimal solution, whereby linearization errors tend to become larger if the linearization point is further away from the real solution. A possibility to improve this aspect is to make use of an IEKF which is basically the recursive form of the Gauss-Newton optimization ([5]).

A nonlinear discrete time system with state $\boldsymbol{x}$, innovation term $\boldsymbol{y}$, process noise $\boldsymbol{w} \sim \mathcal{N}(0, \boldsymbol{W})$, and update noise $\boldsymbol{n} \sim \mathcal{N}(0, \boldsymbol{R})$ can be written as:

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{w}_{k-1}), \tag{11.37}$$

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k, \boldsymbol{n}_k). \tag{11.38}$$

In eq. (11.38) we made use of an implicit formulation of the measurement model which directly yields the Kalman innovation $\boldsymbol{y}_k$. This provides more flexibility in the design by allowing the direct integration of residuals. Given an a-posteriori estimate $\boldsymbol{x}^+_{k-1}$ with covariance $\boldsymbol{P}^+_{k-1}$, the prediction step of the IEKF is analogous to the EKF and yields the a-priori estimate at the next time step:

$$\boldsymbol{x}^-_k = f(\boldsymbol{x}^+_{k-1}, \boldsymbol{0}), \tag{11.39}$$

$$\boldsymbol{P}^-_k = \boldsymbol{F}_{k-1}\boldsymbol{P}^+_{k-1}\boldsymbol{F}^T_{k-1} + \boldsymbol{G}_{k-1}\boldsymbol{W}_{k-1}\boldsymbol{G}^T_{k-1}, \tag{11.40}$$

with the Jacobians

$$\boldsymbol{F}_{k-1} = \frac{\partial f}{\partial \boldsymbol{x}_{k-1}}(\boldsymbol{x}^+_{k-1}, \boldsymbol{0}), \tag{11.41}$$

$$\boldsymbol{G}_{k-1} = \frac{\partial f}{\partial \boldsymbol{w}_{k-1}}(\boldsymbol{x}^+_{k-1}, \boldsymbol{0}). \tag{11.42}$$

Analogously to the EKF, the update step of the IEKF can be linked to an optimization problem considering the deviation from the prior $\boldsymbol{x}_k^-$ and the innovation term $\boldsymbol{h}(\boldsymbol{x}_k^+, \boldsymbol{0})$:

$$\min_{\boldsymbol{x}_k^+} \|\boldsymbol{x}_k^+ \boxminus \boldsymbol{x}_k^-\|_{\boldsymbol{P}_k^{-1}} + \|\boldsymbol{h}(\boldsymbol{x}_k^+, \boldsymbol{0})\|_{(\boldsymbol{J}_k \boldsymbol{R}_k \boldsymbol{J}_k^T)^{-1}}. \tag{11.43}$$

However, in contrast to the EKF, an iterative scheme is employed where the problem is linearized around continuously refined linearization points $\boldsymbol{x}_{k,j}^+$ starting with $\boldsymbol{x}_{k,0}^+ = \boldsymbol{x}_k^-$:

$$\begin{aligned}\min_{\Delta\boldsymbol{x}_{k,j}} &\|\boldsymbol{x}_{k,j}^+ \boxminus \boldsymbol{x}_k^- + \boldsymbol{L}_{k,j}^{-1}\Delta\boldsymbol{x}_{k,j}\|_{\boldsymbol{P}_k^{-1}} \\ &+ \|\boldsymbol{h}(\boldsymbol{x}_{k,j}^+, \boldsymbol{0}) + \boldsymbol{H}_{k,j}\Delta\boldsymbol{x}_{k,j}\|_{(\boldsymbol{J}_k \boldsymbol{R}_k \boldsymbol{J}_k^T)^{-1}}\end{aligned} \tag{11.44}$$

where the Jacobians are updated every iteration step:

$$\boldsymbol{H}_{k,j} = \frac{\partial h}{\partial \boldsymbol{x}_k}(\boldsymbol{x}_{k,j}^+, \boldsymbol{0}), \tag{11.45}$$

$$\boldsymbol{J}_{k,j} = \frac{\partial h}{\partial \boldsymbol{n}_k}(\boldsymbol{x}_{k,j}^+, \boldsymbol{0}), \tag{11.46}$$

$$\boldsymbol{L}_{k,j} = \frac{\partial(\boldsymbol{x}_k^- \boxplus \Delta\boldsymbol{x})}{\partial\Delta\boldsymbol{x}}(\boldsymbol{x}_{k,j}^+ \boxminus \boldsymbol{x}_k^-). \tag{11.47}$$

The Jacobian $\boldsymbol{L}_{k,j}$ of the boxplus operator is required to account for the special linearization of certain states such as rotations or bearing vectors. Its inverse $\boldsymbol{L}_{k,j}^{-1}$ is the corresponding Jacobian of the boxminus operation w.r.t. to the left operand and is required to linearize the deviation of the prior in (11.44). Please note that due to the special notion of differentials on manifolds the Jacobian $\boldsymbol{L}_{k,j}$ is a square matrix (see eq. (11.11)). Also, in the case of pure vector spaces this Jacobian will be the identity matrix.

Setting the derivative of the cost function (11.44) w.r.t. the incremental update $\Delta\boldsymbol{x}_{k,j}$ to zero and employing some matrix calculus yields the following recursive solution:

$$\boldsymbol{S}_{k,j} = \boldsymbol{H}_{k,j}\boldsymbol{L}_{k,j}^T\boldsymbol{P}_k^-\boldsymbol{L}_{k,j}\boldsymbol{H}_{k,j}^T + \boldsymbol{J}_{k,j}\boldsymbol{R}_k\boldsymbol{J}_{k,j}^T, \tag{11.48}$$

$$\boldsymbol{K}_{k,j} = \boldsymbol{L}_{k,j}^T\boldsymbol{P}_k^-\boldsymbol{L}_{k,j}\boldsymbol{H}_{k,j}^T\boldsymbol{S}_{k,j}^{-1}, \tag{11.49}$$

$$\begin{aligned}\Delta\boldsymbol{x}_{k,j} = \boldsymbol{K}_{k,j}\Big(&\boldsymbol{H}_{k,j}\boldsymbol{L}_{k,j}(\boldsymbol{x}_{k,j}^+ \boxminus \boldsymbol{x}_k^-) \\ &- \boldsymbol{h}(\boldsymbol{x}_{k,j}^+, \boldsymbol{0})\Big) - \boldsymbol{L}_{k,j}(\boldsymbol{x}_{k,j}^+ \boxminus \boldsymbol{x}_k^-),\end{aligned} \tag{11.50}$$

$$\boldsymbol{x}_{k,j+1}^+ = \boldsymbol{x}_{k,j}^+ \boxplus \Delta\boldsymbol{x}_{k,j}, \tag{11.51}$$

whereby the iteration is terminated when the update $\Delta\boldsymbol{x}_{k,j}$ is below a certain threshold. Finally, the covariance matrix is only updated once the process has converged

after n iterations:

$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_{k,n}\boldsymbol{H}_{k,n})\boldsymbol{L}_{k,n}^T\boldsymbol{P}_k^-\boldsymbol{L}_{k,n}. \tag{11.52}$$

Especially in setups with large initial uncertainties, the IEKF can provide a significant gain in convergence and accuracy. Using a termination criteria based on the magnitude of the performed correction, the computational overhead can be limited to cases with large update corrections (e.g. the initial measurements of a newly included landmark). Once the state has properly converged, the number of iterations can be kept to a minimum and the computational effort remains similar to the regular EKF.

5.2 Filter Setup and State Definition

Similar to other visual-inertial filtering frameworks, the inertial measurements are employed to propagate the filter state, while the visual measurements are processed and integrated during the filter update step ([73, 79]). The proposed filter setup differs in that it makes use of a fully robocentric formulation of the filter state, which has previously been tested in vision-only approaches [25]. The advantage of this formulation is that the position and yaw states, which are unobservable, can be fully decoupled from the rest of the filter states. This decreases the noise magnitude and improves the consistency of relevant states such as velocity or inclination angles. On the other hand, noise from the gyroscope affects all states that need to be rotated during the state propagation (see section 5.3). However, since the gyroscope noise is often relatively small and because most states are observable, this does not represent a significant issue.

The state of the filter is composed of the following elements (including N visual landmarks):

$$\boldsymbol{x} := \big(\boldsymbol{r}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{b}_f, \boldsymbol{b}_\omega, \boldsymbol{c}, \boldsymbol{z}, \boldsymbol{\mu}_0, \ldots, \boldsymbol{\mu}_N, \rho_0, \ldots, \rho_N\big) \tag{11.53}$$

with:

- $\boldsymbol{r} := {}_{\mathcal{B}}\boldsymbol{r}_{\mathcal{IB}}$: robocentric position of IMU,
- $\boldsymbol{v} := {}_{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}}$: robocentric velocity of IMU,
- $\boldsymbol{q} := \boldsymbol{q}_{\mathcal{IB}}$: attitude of IMU (map from $\mathcal{B}$ to $\mathcal{I}$),
- $\boldsymbol{b}_f$: additive bias on accelerometer (expressed in $\mathcal{B}$),
- $\boldsymbol{b}_\omega$: additive bias on gyroscope (expressed in $\mathcal{B}$),
- $\boldsymbol{c} := {}_{\mathcal{B}}\boldsymbol{r}_{\mathcal{BC}}$: linear part of IMU-camera extrinsics,
- $\boldsymbol{z} := \boldsymbol{q}_{\mathcal{CB}}$: rotational part of IMU-camera extrinsics,
- $\boldsymbol{\mu}_i$: bearing vector to landmark i (expressed in $\mathcal{C}$),
- ρ_i: distance parameter of landmark i.

The generic parametrization for the distance d_i of a landmark i is given by the mapping $d_i = d(\rho_i)$ (with derivative $d'(\rho_i)$). In the context of this work we mainly tested the inverse distance parametrization, $d(\rho_i) = 1/\rho_i$. A brief comparison with the regular distance parametrization is provided in section 7.2.

Rotations $(\boldsymbol{q}, \boldsymbol{z})$ and bearing vectors $(\boldsymbol{\mu}_i)$ are parametrized as detailed in section 3.2 and section 3.3. This means that quantities like differences, uncertainties, or errors are represented as elements of a vector space with minimal dimension, i.e., 3D for rotations and 2D for bearing vectors. By using the combined bearing vector and distance parametrization, landmarks can be initialized in an *undelayed* manner and can be integrated into the filter at detection time. The distance of a landmark is initialized with a fixed value or, if sufficiently converged, with an estimate of the current average scene distance. The corresponding covariance is set to a very large value. In comparison to other parametrizations, we do not over-parametrize the 3D landmark location estimates, whereby each landmark corresponds to 3 columns in the covariance matrix of the state (2 for the bearing vector and 1 for the distance parameter). This also avoids the need for re-parametrization ([126]).

A singularity-free parametrization of bearing vectors on the full unit sphere is essential here. It enables the proper representation of bearing vectors and their uncertainty estimates even if outside the field of view of the camera. Furthermore, limiting the validity of the parametrization to a certain region would render online camera-IMU extrinsics calibration and multi-camera support more difficult.

5.3 State Propagation

The state propagation is driven by the proper acceleration measurement, $\tilde{\boldsymbol{f}} = {}_{\mathcal{B}}\tilde{\boldsymbol{f}}_{\mathcal{B}}$, and the rotational rate measurement, $\tilde{\boldsymbol{\omega}} = {}_{\mathcal{B}}\tilde{\boldsymbol{\omega}}_{\mathcal{I}\mathcal{B}}$. Both measurements are modeled as noise and bias affected leading to the following bias corrected but noise affected estimates:

$$\hat{\boldsymbol{f}} = \tilde{\boldsymbol{f}} - \boldsymbol{b}_f - \boldsymbol{w}_f, \tag{11.54}$$

$$\hat{\boldsymbol{\omega}} = \tilde{\boldsymbol{\omega}} - \boldsymbol{b}_\omega - \boldsymbol{w}_\omega. \tag{11.55}$$

Together with the estimated camera linear velocity and rotational rate

$$\hat{\boldsymbol{v}}_{\mathcal{C}} = \boldsymbol{z}(\boldsymbol{v} + \hat{\boldsymbol{\omega}}^{\times}\boldsymbol{c}), \tag{11.56}$$

$$\hat{\boldsymbol{\omega}}_{\mathcal{C}} = \boldsymbol{z}(\hat{\boldsymbol{\omega}}), \tag{11.57}$$

this yields the following set of continuous differential equations:

$$\dot{\boldsymbol{r}} = -\hat{\boldsymbol{\omega}}^{\times}\boldsymbol{r} + \boldsymbol{v} + \boldsymbol{w}_r, \tag{11.58}$$

$$\dot{\boldsymbol{v}} = -\hat{\boldsymbol{\omega}}^{\times}\boldsymbol{v} + \hat{\boldsymbol{f}} + \boldsymbol{q}^{-1}(\boldsymbol{g}), \tag{11.59}$$

$$\dot{\boldsymbol{q}} = -\boldsymbol{q}(\hat{\boldsymbol{\omega}}), \tag{11.60}$$

$$\dot{\boldsymbol{b}}_f = \boldsymbol{w}_{bf}, \tag{11.61}$$

$$\dot{\boldsymbol{b}}_\omega = \boldsymbol{w}_{b\omega}, \tag{11.62}$$

$$\dot{\boldsymbol{c}} = \boldsymbol{w}_c, \tag{11.63}$$

$$\dot{\boldsymbol{z}} = \boldsymbol{w}_z, \tag{11.64}$$

$$\dot{\boldsymbol{\mu}}_i = \boldsymbol{N}(\boldsymbol{\mu}_i)^T \left(\hat{\boldsymbol{\omega}}_{\mathcal{C}} + \boldsymbol{n}(\boldsymbol{\mu}_i)^{\times} \frac{\hat{\boldsymbol{v}}_{\mathcal{C}}}{d(\rho_i)} \right) + \boldsymbol{w}_{\mu,i}, \tag{11.65}$$

$$\dot{\rho}_i = -\boldsymbol{n}(\boldsymbol{\mu}_i)^T \hat{\boldsymbol{v}}_{\mathcal{C}} / d'(\rho_i) + w_{\rho,i}. \tag{11.66}$$

The term $\boldsymbol{N}(\boldsymbol{\mu}_i)^T$ projects a 3D vector onto the 2D tangent space at the bearing vector $\boldsymbol{\mu}_i$ (see Figure 11.1). Furthermore, $\boldsymbol{g}$ is the gravity vector expressed in the world coordinate frame, and the terms of the form $\boldsymbol{w}_*$ are white Gaussian noise processes. The corresponding covariance parameters can either be derived from the IMU specifications or can be tuned manually.

Note that the derivatives of bearing vectors and rotations lie within 2D and 3D vector spaces, respectively. This is required for achieving a minimal and consistent representation of the filter state and covariance. While most of the above derivatives are relatively well known, the dynamics of the bearing vector and distance parameter is a novelty in this work. We give a sketch of the corresponding derivation in section 9. It relies on the assumption that a 3D point landmark $\mathcal{F}$ remains stationary with respect to an inertial frame $\mathcal{I}$:

$${}_{\mathcal{I}}\boldsymbol{r}_{\mathcal{I}\mathcal{F}} = {}_{\mathcal{I}}\boldsymbol{r}_{\mathcal{I}\mathcal{C}} + \boldsymbol{q}_{\mathcal{C}\mathcal{I}}^{-1}(\boldsymbol{\mu}\, d(\rho)). \tag{11.67}$$

In eq. (11.66) we can observe that the derivative of the distance parameter only depends on the velocity in direction of the bearing vector. On the other hand, the derivative of the bearing vector, eq. (11.65), is the sum of a velocity and rotational rate effect, whereby the magnitude of the velocity effect is proportional to the inverse distance of the specific landmark.

Using an appropriate Euler forward integration scheme, i.e., using the boxplus operator where appropriate, the above time continuous equation can be transformed into a set of discrete prediction equations which are used during the prediction of the filter state. For the attitude, the rotational IMU-camera extrinsics and the bearing vectors the discretization yields:

$$\begin{aligned}
\boldsymbol{q}_{k+1} &= \boldsymbol{q}_k \boxplus (-\Delta t\, \boldsymbol{q}_k(\hat{\boldsymbol{\omega}}_k)), \\
&= \exp(-\Delta t\, \boldsymbol{q}_k(\hat{\boldsymbol{\omega}}_k)) \otimes \boldsymbol{q}_k, \\
&= \boldsymbol{q}_k \otimes \exp(-\Delta t\, \hat{\boldsymbol{\omega}}_k) \otimes \boldsymbol{q}_k^{-1} \otimes \boldsymbol{q}_k, \\
&= \boldsymbol{q}_k \otimes \exp(\Delta t(\boldsymbol{b}_{\omega,k} + \boldsymbol{w}_{\omega,k} - \tilde{\boldsymbol{\omega}}_k)),
\end{aligned} \tag{11.68}$$

$$\begin{aligned}
\boldsymbol{z}_{k+1} &= \boldsymbol{z}_k \boxplus \Delta t \boldsymbol{w}_{z,k}, \\
&= \exp(\Delta t \boldsymbol{w}_{z,k}) \otimes \boldsymbol{z}_k,
\end{aligned} \tag{11.69}$$

$$\begin{aligned}
\boldsymbol{\mu}_{i,k+1} = \exp\Big(\Delta t \Big(& \big(\boldsymbol{I} - \boldsymbol{n}(\boldsymbol{\mu}_{i,k}) \boldsymbol{n}(\boldsymbol{\mu}_{i,k})^T\big)\, \hat{\boldsymbol{\omega}}_{\mathcal{C}} \\
& + \boldsymbol{n}(\boldsymbol{\mu}_{i,k})^{\times} \frac{\hat{\boldsymbol{v}}_{\mathcal{C}}}{d(\rho_{i,k})} + \boldsymbol{N}(\boldsymbol{\mu}_{i,k}) \boldsymbol{w}_{\mu,i} \Big) \Big) \otimes \boldsymbol{\mu}_{i,k}.
\end{aligned} \tag{11.70}$$

The derivation of the bearing vector discretization is given in section 9.

In typical visual-inertial sensor setups, the IMU measurements are often obtained at a higher rate than the images. As the proposed propagation step is driven by the IMU measurements this can result in a high computational burden. A classical approach to mitigate this issue is to make use of IMU pre-integration techniques ([41]) in order to merge multiple IMU measurements into a single prediction step. However, since the duration between two consecutive images remains relatively small we employ a simpler pre-integration approach where the Jacobian is evaluated based on the mean of the IMU measurement. Thus, even if multiple IMU measurements are available between two consecutive images, eq. (11.40) is evaluated only once. Compared to the regular solution no notable performance loss could be observed.

5.4 Direct Innovation Term and Update

In section 4.3, we discussed how to construct a photometric error term based on the pixel-wise intensity difference between a previously extracted patch and its predicted location in a given image frame. Within an IEKF this can be directly used as innovation term. However, for the multilevel patch format that we use, this would lead to a $6 \times 6 \times 2 = 72$ dimensional error term per patch inducing very high computational cost. Fortunately, looking at eq. (11.34), one can observe that the entire error term corresponding to a patch P_i and an image I is only dependent on the estimated pixel coordinates $\boldsymbol{p}_i = \boldsymbol{\pi}(\boldsymbol{\mu}_i)$. Thus, the only direct filter state dependency of this error term is given by the bearing vector and an equivalent reduced 2D error term can be derived. This can be achieved by means of a QR-decomposition of the gradient matrix in eq. (11.34):

$$\begin{aligned} \boldsymbol{A}(\boldsymbol{p}_i, I, \boldsymbol{D}_i) &= \boldsymbol{Q}(\boldsymbol{p}_i, I, \boldsymbol{D}_i)\boldsymbol{R}(\boldsymbol{p}_i, I, \boldsymbol{D}_i), \\ &= \begin{bmatrix}\boldsymbol{Q}_1(\boldsymbol{p}_i, I, \boldsymbol{D}_i) & \boldsymbol{Q}_2(\boldsymbol{p}_i, I, \boldsymbol{D}_i)\end{bmatrix} \begin{bmatrix}\boldsymbol{R}_1(\boldsymbol{p}_i, I, \boldsymbol{D}_i) \\ \boldsymbol{0}\end{bmatrix} \end{aligned} \tag{11.71}$$

where the upper-triangular matrix $\boldsymbol{R}_1(\boldsymbol{p}_i, I, \boldsymbol{D}_i)$ has full row-rank 2 for regular features, row-rank 1 for line features, and goes towards $\boldsymbol{0}$ for uniform patches.

Considering the above decomposition, the innovation term for the j^{th} iteration step for a patch i yields:

$$\boldsymbol{y}_{i,j} = \boldsymbol{Q}_1(\boldsymbol{\pi}(\boldsymbol{\mu}_{i,j}^+), I, \boldsymbol{D}_i)^T \boldsymbol{b}(\boldsymbol{\pi}(\boldsymbol{\mu}_{i,j}^+), P_i, I, \boldsymbol{D}_i). \tag{11.72}$$

This has a maximal dimension of 2 and loses dimensions for degenerate cases like line-shaped or uniform patches. Since this represents a left-multiplication with an orthonormal matrix, the noise characteristics are assumed to be of the same magnitude on every single error term. To account for potentially different noise properties of the intensity errors, a weighting based scheme could be introduced. The Jacobian for the innovation term is given by

$$\boldsymbol{H}_{i,j} = \boldsymbol{R}_1(\boldsymbol{\pi}(\boldsymbol{\mu}_{i,j}^+), I, \boldsymbol{D}_i) \frac{d\boldsymbol{\pi}}{d\boldsymbol{\mu}}(\boldsymbol{\mu}_{i,j}^+). \tag{11.73}$$

Within the IEKF, the tracked landmarks are updated one after another, each undergoing a certain number of iterations. While the robocentric state formulation

moved parts of the nonlinearities from the update into the propagation step, significant nonlinearities remain with the pixel intensity generation. The update iterations are taking care of aligning the patches in the current image and, simultaneously, to spread the gained information throughout the filter state. Thus, all the landmark detection and tracking functionality is intrinsically contained in the filter. In the case where a landmark's predicted image coordinates exhibit a large uncertainty (e.g. for newly initialized landmarks), multiple hypothesis are select within the uncertainty bound. Figure 11.4 provides a simplified sketch of the tracking concept. An advantage of this is that non-corner features can be properly tracked by considering the prior provided by the IMU-driven process model. In the case of line-shaped features, for instance, a corrective update only applies along the perpendicular direction to the line, while the other direction remains unaffected. In the degenerate case of uniformly textured patch features, the iteration finishes after one step without changes to the filter state (since no information is contained in the patch). Figure 11.5 shows the tracked landmarks in a frame. Each iteration for a landmark update is depicted by a yellow dot. Especially for the newly added landmarks (the four most right ones), the initial uncertainty (yellow) ellipse and the number of iterations are increased.

To account for moving objects or other disturbances, a simple Mahalanobis based outlier detection is implemented within the update step. It compares the obtained innovation residual with the predicted innovation covariance and rejects the measurement whenever the weighted norm exceeds a certain threshold. This method inherently takes into account the covariance of the state and measurements. It also considers the image gradients and thereby tends to reject gradient-less image patches more readily. In addition, a threshold on the total intensity error of a patch is introduced, whereas a patch measurement is rejected if the threshold is exceeded. Also, a landmark quality check is performed by sampling 4 nearby locations and evaluating the corresponding innovation residual. Tracking tends to be bad if not at least two locations exhibit a significantly higher residual than the matched landmark (see the bottom left landmark in Figure 11.5).

The computation of the photometric error relies on an image patch from a previous frame. If parts of this image patch have influenced the filter state in the past, the resulting photometric error will exhibit a correlation with the current filter state. This correlation is not modeled in the current framework and doing so would significantly increase the computational burden (one possible approach would be to co-estimate the patch pixel intensities). This is an issue which is also commonly encountered in dense approaches where cross-correlations between localization and mapping are often neglected. In our case however, the cross-correlation with the environment geometry is tracked and accounted for and the problem is limited to the texture of the environment. A refinement step on the patch intensities could reduce the pixel intensity noise and thereby reduce this effect. Investigations in this direction will be part of future work.

5.5 Landmark Management

The IEKF does not exhibit good scalability in terms of the size of the filter state. Consequently, only a limited number of landmarks can be tracked and they have to be selected and managed carefully in order to obtain good results. In section 4.4,

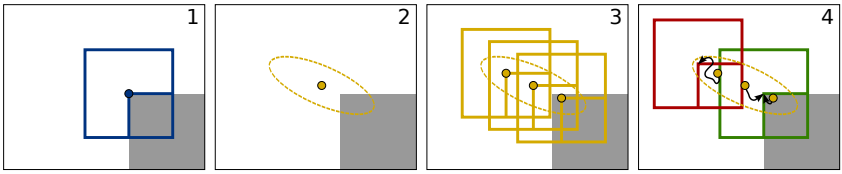

Figure 11.4: Overview of the landmark tracking concept. Step 1: a patch feature (blue square) is extracted for the landmark (blue dot). Step 2: the estimated landmark image coordinates (yellow dot) and the corresponding covariance (yellow ellipse) are provided by the filter's IMU-driven process model. Step 3: depending on the magnitude of the uncertainty multiple candidates (yellow dots) are initialized. Step 4: for each candidate an iterative update (black arrow) is performed which integrates patch intensity errors together with the motion prior. Outlier detection and quality checks are performed to select the best valid tracking (green vs red squares). Steps 3 and 4 are completely integrated in the iterative filter updates.

we outlined an intensity based scoring which describes how informative the content of a patch can be. This is mainly used to decide what landmarks are added to the filter state. In addition to this, we maintain tracking and visibility information of a landmark, and a combined heuristic quality score is computed for each landmark which is being tracked. The quality score is composed of three sub-scores:

- The global quality: how often has a landmark been tracked since initialization
- The local quality: how often has a landmark been tracked when expected to be in the field of view (limited to recent frames)
- The local visibility: how often was the landmark in the field of view (limited to recent frames)

If a landmark exhibits a high global quality, i.e. it has often been tracked since initialization, the pruning threshold on the two local sub-scores is kept more conservative. Using an adaptive thresholding, we can control the total amount of landmarks which are currently in the frame. E.g. if we reach the maximal number of landmarks in the filter state and only a minor part is properly tracked, we make the landmark pruning stricter to get space for new landmarks.

6 Multi-Camera Setup

One issue with monocular visual-inertial setups is that they require sufficient motion in order to properly estimate the complete filter state. Also, a particular camera can be blind at times, because of fast lightning changes or very bad texture. Adding an additional camera can therefore improve the robustness of the overall system. In the case where a multi-camera setup has overlapping fields of view, multiple measurements

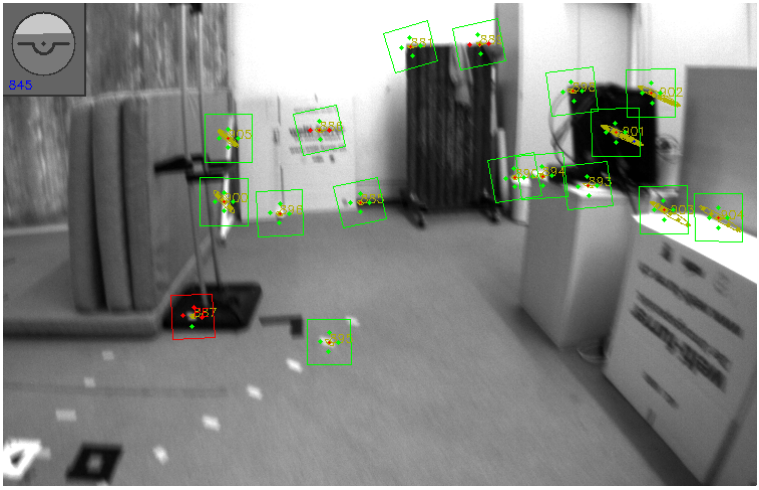

Figure 11.5: Live screenshot of the tracked landmarks. The projected patches (final iteration) are depicted by squares (green if successful, red if rejected). The predicted uncertainty of the landmark location are represented by yellow ellipses and each update iteration candidates is marked by a yellow dot. The final location is highlighted with a small red dot surrounded by four green or red dots. The surrounding locations are checked for higher innovation residuals (green). If more than two surrounding locations exhibit no increased innovation residuals (red) then the match is rejected (e.g. the bottom left landmark).

of the same landmark are received at a given time. This provides information about the landmark's distance and the extrinsic calibration of the corresponding camera frames. Still, some excitation of the states is necessary to estimate the calibration parameters. Once the calibration estimates have converged, the distance of landmarks in overlapping fields of view becomes observable, even if the sensor remains stationary.

New landmarks are still detected in single camera frames only and the parametrization of the corresponding bearing vector and distance parameter is kept with respect to the detection frame. In the case where the newly detected landmark can be seen in more than one camera frame, the initial distance estimate can be computed by triangulation.

For all subsequent time steps, the landmarks get projected in every camera frame. If the predicted pixel coordinates lie within a camera frame, an iterative update is performed (see section 5.4). If the measurement camera frame $\mathcal{C}_m$ (where the landmark is observed) is not the same as the detection camera frame $\mathcal{C}_d$ (where the landmark is parametrized), the corresponding bearing vector must be transformed into the mea-

surement frame first. This can be done by

$$ {}_{\mathcal{C}_m}\boldsymbol{\mu}_i = \boldsymbol{z}_m(\boldsymbol{c}_d + \boldsymbol{z}_d^{-1}({}_{\mathcal{C}_d}\boldsymbol{\mu}_i\, d(\rho_i)) - \boldsymbol{c}_m) \tag{11.74} $$

where the terms of the form $\boldsymbol{c}_*$ and $\boldsymbol{z}_*$ represent the linear and rotational extrinsic IMU-camera calibration of the corresponding camera. Together with the parametrization of the landmark location $({}_{\mathcal{C}_d}\boldsymbol{\mu}_i, \rho_i)$, they are contained in the filter state. This represents the main difference to the monocular setup, whereas the innovation Jacobian in eq. (11.73) has to be right-multiplied by the Jacobian of eq. (11.74).

7 Experimental Results

The evaluation of the presented approach and its implementation, Rovio, is split into three parts. Whenever possible, the publicly available EuRoC datasets are employed ([20]). In the first set of experiments, the filter convergence is analyzed. One of the most critical quantities in this respect is the estimated distance of the landmarks which is unknown at initialization. Having good convergence properties is crucial for the performance of the filter. The convergence of different online calibration parameters such as IMU-camera extrinsics and IMU biases is also analyzed.

In a second part, we investigate the performance of the approach in terms of accuracy. We compare different filter parameters with each other and evaluate the results against the state-of-the-art visual-inertial odometry framework Okvis ([84]). Basic comparison with the frameworks of [41] and [136] is also provided. Finally, results are presented where Rovio is used within a feedback control loop on a MAV (Micro Aerial Vehicle) performing aggressive maneuvers. In this context, we investigate the control critical estimates such as velocity and inclination angles. This also includes an analysis of the estimated covariances.

All experiments are performed with the same selection of filter parameters except where explicitly mentioned. Our baseline implementation only employs the second and third image pyramid level. While taking into account the first image level can increase accuracy, it is not really useful for highly dynamic and difficult cases. The same parameters are selected for the monocular and stereo setup. We use a patch size of 6×6 together with 25 filter landmarks. Some parameter variations are investigated in section 7.2.

7.1 Convergence Evaluation

As landmarks are initialized with a very high distance uncertainty, the convergence of the corresponding covariance is a critical aspect. While a decreasing uncertainty is desired since it allows for more accurate tracking of the sensor pose, spurious and inconsistent convergence must be avoided. Especially in the monocular case, the uncertainty should only be decreased if the sensor is moving and sufficient baseline is acquired. In order to investigate the consistency of the distance estimates, a dataset was recorded where a horizontal surface was the only visible object (the camera was directed towards the floor). The virtual groundtruth of the sensor height is inferred by averaging over the height of all converged landmarks (see Figure 11.6). This allows

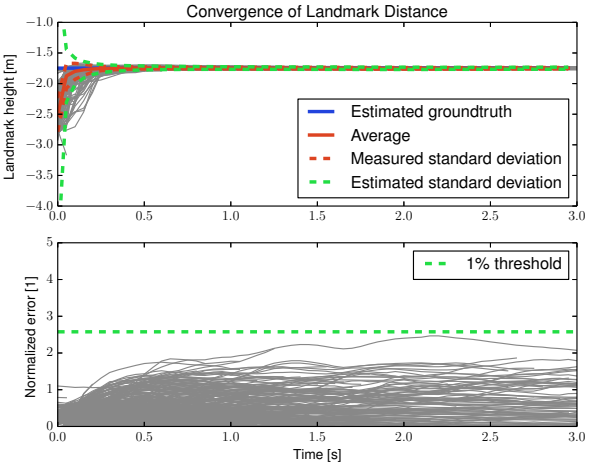

Figure 11.6: Dataset observing a horizontal plane with a *monocular* setup. Estimated landmark heights (grey) together with their average (red) and empirical standard deviation (red dashes, 1-σ). The groundtruth (blue) is estimated by averaging the height estimates once converged. In the top plot, the average of the estimated standard deviation over multiple landmark heights is provided (green dashes, 1-σ). The bottom plot depicts the normalized error (height error divided by estimated standard deviation) together with the 1% confidence threshold.

to evaluate the convergence of the landmarks heights (which are strongly coupled to the distance estimates).

Figures 11.6 and 11.7 show the estimated height of the tracked landmarks over time. Since the landmarks are initialized at a fixed distance, which in this experiment tends to relate to points below the surface, a significant estimation error can be observed at initialization. Due to the motion of the sensor, however, the height estimates quickly converge. In the top part of both figures, the estimated standard deviation (average of the estimated standard deviations) is compared against the measured standard deviation (empirical standard deviation of the actual height errors). In both cases, the estimated standard deviation encompasses the measured one. Since we averaged over many landmark tracks, this is not a strict check of consistency but still shows that the estimated covariance must lie within a reasonable range on average. A better analysis is provided in both lower plots which depict the normalized height error (height error divided by estimated standard deviation). In both experiments we can show that the normalized error remains below the 1% confidence threshold, i.e., there are no unreasonably large height estimate errors if compared to the corresponding

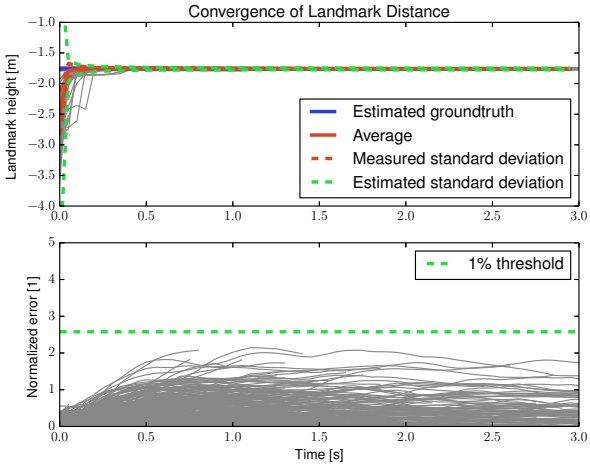

Figure 11.7: Dataset observing a horizontal plane with a *stereo* setup. Stereo initialization is disabled. Estimated landmark heights (grey) together with their average (red) and empirical standard deviation (red dashes, 1-σ). The groundtruth (blue) is estimated by averaging the height estimates once converged. In the top plot, the average of the estimated standard deviation over multiple landmark heights is provided (green dashes, 1-σ). The bottom plot depicts the normalized error together with the 1% confidence threshold.

uncertainty estimates.

The results of the monocular and the stereo setup are similar due to the significant amount of motion present in the recorded data. The final standard deviation of the height errors amounts to 0.0119 m for the monocular setup and 0.0073 m for the stereo setup.

Other parameters which have to converge for a proper functioning of the filter are the online calibration parameters, which are composed of the IMU biases and the IMU-camera extrinsics. The later should remain nearly constant for different datasets with the same sensor setup, and we can thus evaluate the extrinsics on multiple datasets and compare the values they have converged to. Figure 11.8 shows the final estimate of the rotational and translation part of the extrinsics if running the proposed filter on all 11 EuRoC datasets. To make the task more difficult, the initial values were selected as zero translation and closest orthogonal rotation (corresponding to all zero angles in the figure). The resulting estimates, including uncertainties, seem to exhibit a large amount of accordance between the different datasets and between monocular and stereo setup. In comparison to the first half of the datasets, the second half includes datasets with less motion which pose more difficulties for a proper estimation of the

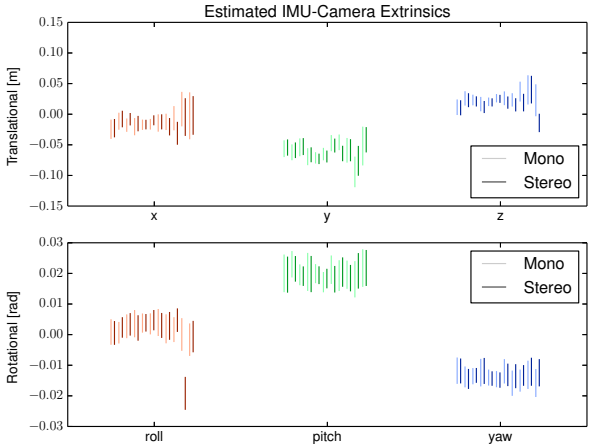

Figure 11.8: Final camera-IMU extrinsics estimates for all 11 EuRoC datasets for monocular and stereo setups. The length of the lines corresponds to the 3-σ bounds of the estimates. The order of the lines corresponds to the datasets V1_01–03, V2_01–03, and MH_01–05.

extrinsics. Consequently, the estimated uncertainty (length of bar) remains larger as well. In general, the stereo setup only brings a marginal reduction of the uncertainties as long as sufficient motion is available. On the contrary, the stereo setup exhibits more difficulties for converging to the proper extrinsic calibration, as can be seen in the lower plot where a single roll angle converges to a biased value. This is due to wrong stereo matches, whereby a single wrong match can bias the estimation, especially if there is not sufficient motion for correcting the wrong convergence.

For completeness, we also investigate the convergence of the IMU bias estimation. We only evaluate them on a single dataset (V1_03) since they exhibit intra-dataset variability. Since we have a stereo setup, we can perform two distinct monocular as well as one stereo evaluation with the same dataset. The results are depicted in Figures 11.9 and 11.10, where the estimate over time is plotted together with the 3-σ bounds. In particular, the gyroscope biases seem to converge very rapidly and exhibit only a very small variability. Also the accelerometer biases converge with sufficient excitation of the system. They typically converge faster along the gravity direction which is given by the x-axis at the beginning of the dataset.

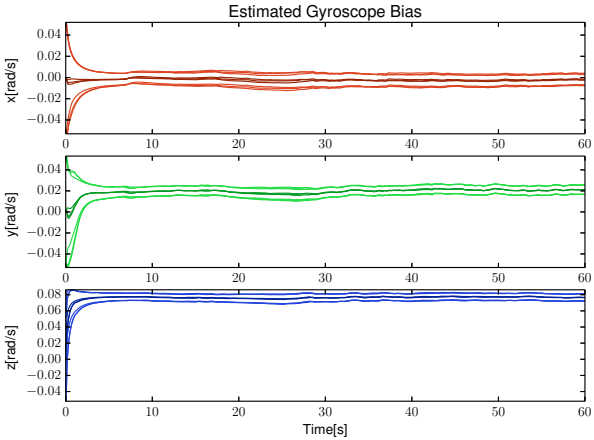

Figure 11.9: Estimated gyroscope biases for dataset V1_03. Estimates (darker lines) together with the 3-σ bounds (brighter lines). Results for two monocular (left and right camera) and one stereo evaluation are depicted. The estimates converge very quickly and are less motion dependent than the accelerometer biases.

7.2 Accuracy Evaluation

Accuracy remains an important criteria for visual-inertial odometry and can be evaluated quantitatively. To a certain extent, it can also serve as a surrogate measure for the well-functioning of an approach. Here we evaluate different parameter setups in order to determine the influence of different aspects of our approach. We also provide a comparison against the state-of-the-art visual-inertial odometry framework Okvis ([84]). Most evaluations are performed on the EuRoC dataset V1_03. To allow comparisons with other visual-inertial frameworks we also provide the results obtained on the long circular dataset which was used by [84], [41], and [136].

To evaluate accuracy we contemplate the root mean square estimation error per traveled distance ([49]). For instance, if we want to evaluate the accuracy after 10 m of traveled distance and have a dataset which is 80 m long, we split the obtained estimation results into 8 chunks of 10 m. The chunks are then aligned with the corresponding bit of groundtruth data and the accumulated error after 10 m is evaluated. Box-plots are employed to depict the corresponding median and quartiles. Assuming that the odometry output exhibits random walk drift with increasing traveled distance (which is often a good approximation as long as the yaw error remains small), the observed errors should increase as square root of the traveled distance. We select the

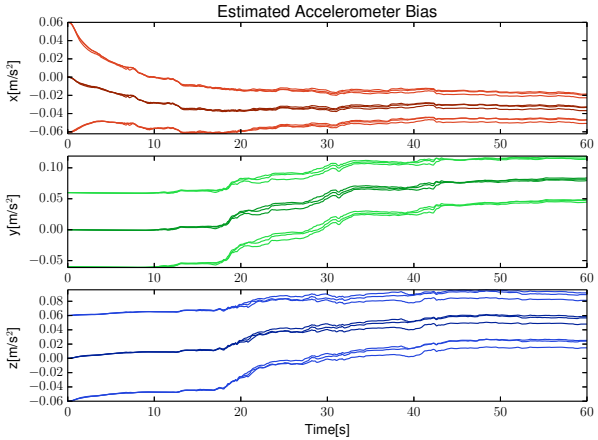

Figure 11.10: Estimated accelerometer biases for dataset V1_03. Estimates (darker lines) together with the 3-σ bounds (brighter lines). Results for two monocular (left and right camera) and one stereo evaluation are depicted. The accelerometer bias converges quicker along the gravity direction which is mostly along the x-axis.

spacing of the traveled distance samples quadratically, and should therefore observe a linear error increase in the plots. The following results can vary depending on the selected dataset and should not be over-interpreted.

Influence of Number of Landmarks and Patch Size

Since the total number of landmarks in the filter state has a major influence on the computational cost of the framework, this is the first parameter that we evaluate. Figure 11.11 shows the position error with respect to the traveled distance for different amounts of landmarks. Surprisingly, increasing the number of landmarks does not improve the accuracy of the output once a certain amount of landmarks is reached (roughly 20). While for vision-only systems it has been shown that the number of landmarks is a crucial parameter ([130]), it seems to be different for visual-inertial systems. In visual-inertial systems the IMU provides a good prior on the motion of the systems and basically needs to be stabilized using recurrent stationary landmarks observations. Consequently, the amount of required landmarks could be depending on the quality of the employed IMU, whereas an IMU of lower quality would benefit more from higher landmark counts. Within this context we also noticed a relatively strong influence of non-rejected outliers on the output's accuracy, whereas we selected the

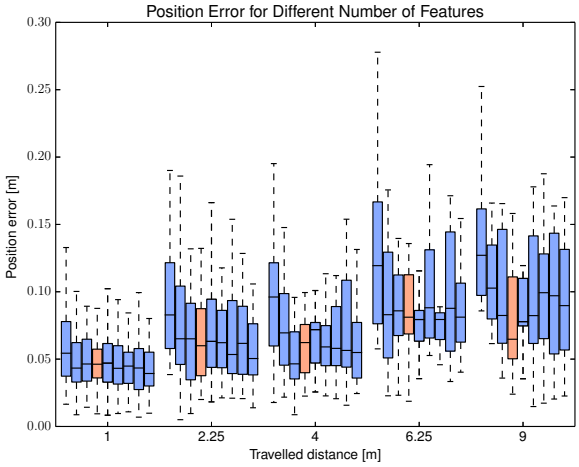

Figure 11.11: Accumulated position error over traveled distance for different landmark counts. The number of landmarks from left to right are 10, 15, 20, 25 (red), 30, 35, 40, 45, 50. The patch size is fixed to 6. The median and the quartiles are depicted.

outlier rejection parameters to be rather strict. We noticed that properly tracking few high-quality landmarks often leads to better results than tracking many landmarks with an increased risk for non-rejected outliers.

In our previous work ([14]) we fixed the patch size to 8×8. Here, we also investigate smaller patch sizes since this reduces the computational load. Results for patch sizes down to 2×2 are depicted in Figure 11.12. It shows that we can reduce the patch size without significantly losing accuracy. Only the case with 2×2 patches exhibits notably increased errors. This could probably be tackled by increasing the amount of pyramid levels which is similar to having larger patches. All in all we propose to employ the combination of patch size 6×6 together with 25 filter landmarks which we highlight in red in all box-plots. On a single core of an Intel® Core™ i7 at 2.4 MHz the resulting framework uses 30-50% of the CPU load.

Effect of Inverse Distance, Photometric Error, and Update Iterations

We investigate the contribution of key components of the proposed framework: the inverse distance parametrization, the update iterations, and the photometric error feedback. In order to assess the effect of inverse distance parametrization we implement the framework with a regular distance parametrization. For the update iterations we limit the update to a single iteration step. The photometric error feedback

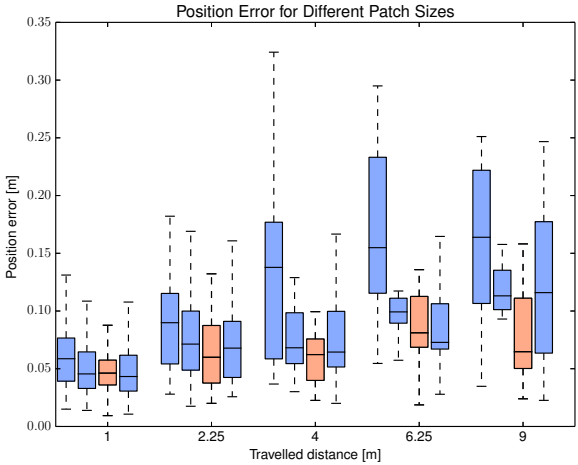

Figure 11.12: Accumulated position error over traveled distance for different patch sizes. The patch sizes from left to right are 2, 4, 6 (red), 8. The landmark count is 25. The median and the quartiles are depicted.

is compared to the traditional reprojection error. This is slightly more involved as a an explicit feature tracker is required. Here we implement the KLT tracker mentioned in section 4.3 (including initial guess from the IMU propagation and using patch warping). In all experiments, the settings are kept as similar as possible. For the reprojection error a different measurement covariance is required, which is tuned to achieve best performance in terms of accumulated position error on the evaluation dataset (V1_03).

The results are depicted in Figure 11.13. The use of a regular distance parametrization leads to significantly increased tracking error. This is due to the less accurate stochastic model on the distance if compared to inverse distance parametrization and confirms previous results ([98]). The EKF implementation exhibits only a slightly increase error metric. This indicates that a single update is often sufficient for proper tracking. This may become more critical if the prediction of the landmark location is less accurate, such as when the initial landmark distance estimate is bad or in cases with high linear velocities. Also the reprojection error based implementation does not lead to much larger tracking error. The reason for this are the abundant corner features in the dataset which are relatively well captured by the regular reprojection error. In the extreme case were no corner features are available (see Figure 11.14) the KLT tracker fails and the advantage of the inherent feature tracking becomes more evident.

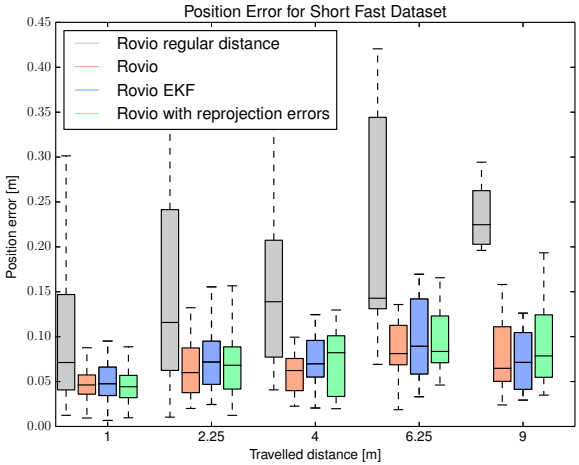

Figure 11.13: Accumulated position error for different Rovio setups. The proposed Rovio setup (red) is compared to an implementation with regular distance parametrization (grey), an implementation without update iterations (blue), and an implementation which uses the reprojection error instead of the photometric error feedback (green). While overall the proposed setup exhibits the smallest tracking error, only the setup with the regular distance parametrization shows a significant accuracy loss.

Comparison with Stereo Setup and Okvis

Figure 11.15 summarizes multiple results. The baseline monocular and stereo Rovio setups are compared against the stereo Okvis framework. When comparing monocular and stereo setups, very similar tracking errors can be observed. This observation can be dataset dependent, but in datasets with a lot of motion (which is the case here), the performance of both setups tends to be very similar. The last set of boxplots corresponds to the results obtained with Okvis, which is the open-source release version of the work of [84]. For this dataset, the performances are comparable and show that our approach can compete with state-of-the-art visual-inertial frameworks.

Finally, the accuracy of the presented approach was also evaluated on the 1.4 km long circular dataset employed by [84], [41], and [136]. Figures 11.16 and 11.17 show the position error and the yaw error over traveled distance for the standard monocular and stereo Rovio setups as well as for Okvis. The performance of Rovio is slightly inferior to Okvis for the 360 m of traveled distance. Again, the performance is strongly depending on the selected tuning parameters which were kept constant for all experiments. [41] and [136] both provide results which show 0.3 m position error after 360 m.

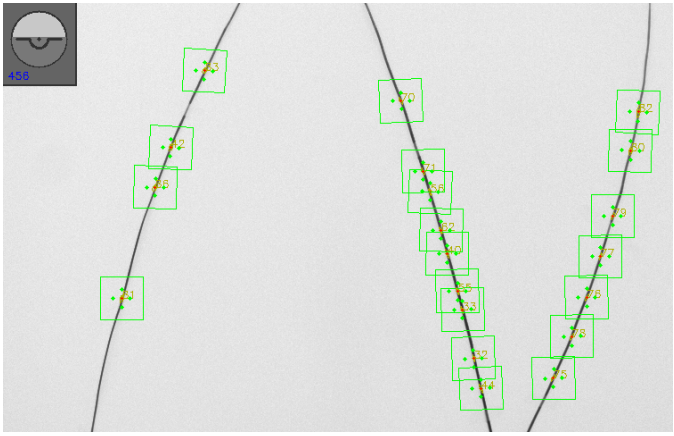

Figure 11.14: Tracking behavior without strong corner features. Snapshot taken at the end of a 20 s dataset with lines only. The inherent feature tracking can handle such situations due to the additional prior it receives from the IMU-driven state propagation.

Caution should be taken when interpreting these results since only 3 non-overlapping segments of length 360 m are contained in the 1.4 km long dataset (low statistical significance). Rovio seems to perform better for shorter distances and shows similar performance to all other visual-inertial frameworks, especially for a traveled distance of 10 m (n=140), where it exhibits a median error of less than 0.1 m. One reason for the decreased long term performance can be found in the increased yaw error which has a strong impact on the position performance for longer distances.

We observed that the above error could be further reduced by choosing stricter outlier rejection parameters and including the first pyramid level into the residual computation (below 0.4 m for 360 m). In the end, however, large scale accuracy should be provided by an enclosing back-end system performing loop closures and re-localization, rather then over-tuning the front-end visual-inertial odometry at the cost of increased computational costs and lower overall robustness.

7.3 Robust MAV Control

In this final evaluation section we investigate the applicability of Rovio for feedback control on a MAV for fast aggressive flights under bad lighting conditions and motion blur. The system is initialized on the ground and remains stationary for 30 s. After take off, it performs three fast circular loops before landing at the same location. The trajectory's position and attitude are depicted in Figures 11.18 and 11.19, respectively. The 3-σ bounds for the estimates are plotted as well. The observable roll and pitch angles very quickly converge from their initially large uncertainties and accurately

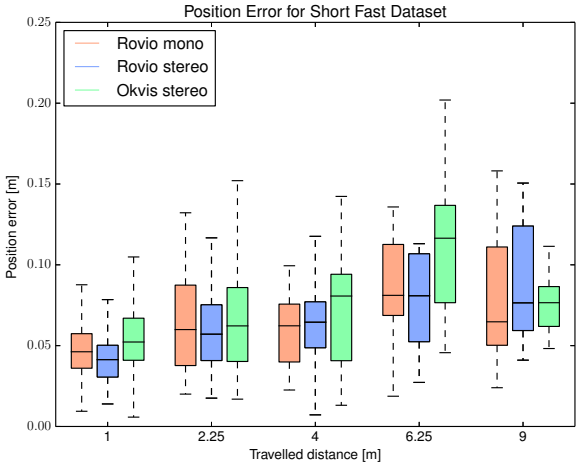

Figure 11.15: Accumulated position error over traveled distance for a monocular and a stereo Rovio setup as well as for Okvis. Red: standard monocular Rovio (6×6 patches, 25 landmarks). Blue: standard stereo Rovio (6×6 patches, 25 landmarks). Green: stereo Okvis with online parameter estimation. All frameworks exhibit similar tracking errors.

track the MAV's inclination angles after take off. The global yaw angle and positions, on the other hand, accumulate uncertainty over time, what confirms the inherent unobservability of those states. All in all, the experiment shows that Rovio can handle fast motions and difficult scenes while providing a reliable state estimation for feedback control of an autonomous MAV.

Figure 11.20 shows the robocentric velocity of the MAV. Due to the robocentric formulation of our filter, the observable states are entirely decoupled from the unobservable states. Hence, the uncertainties are bounded and the estimation error remains minimal which is essential for feedback control. Additionally, velocity estimates are provided where Rovio was reset every 5 s. The estimates very quickly converge to the true velocities for all resets. This highlights the very simple and robust initialization of our robocentric filtering approach where ego-motion estimates are immediately available.

8 Conclusion

This paper presented an IEKF-based framework which tightly fuses inertial measurements and image data. The originality and strength of the proposed approach lie in its

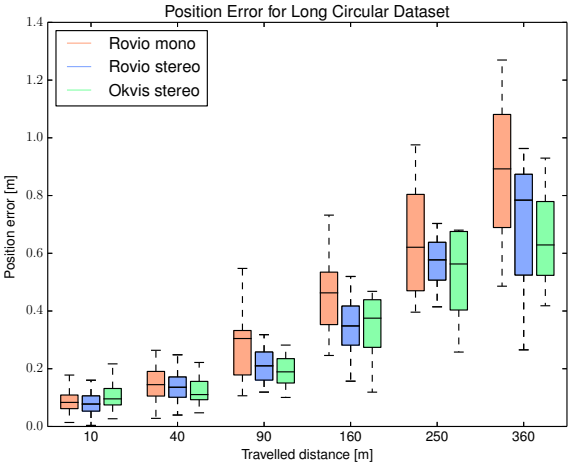

Figure 11.16: Accumulated position error over traveled distance for the long circular dataset. Red: baseline monocular Rovio. Blue: baseline stereo Rovio. Green: stereo Okvis with online parameter estimation.

fully robocentric formulation combined with the direct feedback of photometric error as the Kalman innovation term. This leads to a more robust implementation, since the observable states are not influenced by the growing global covariance. We introduce an iterative update scheme which inherently takes care of landmark tracking. While simplifying the structure of the overall framework, data association is robustified by the tight coupling with the IMU-driven process model. The employed minimal representations of rotations and bearing vectors improve the numerical consistency of the approach and reduce the computational effort. The extensive experimental evaluation shows that the presented approach can compete with state-of-the-art visual-inertial fusion techniques. Interestingly, our approach achieves comparable ego-motion estimation accuracy with a significantly lower landmark count. Robustness with respect to fast motions and bad lightning conditions as well as the instantaneous initialization procedure where demonstrated in a real autonomous MAV flight experiment. Additional features, such as optional GPS measurements, are included in the updated version of the publicly available open-source software package [1].

[1] https://github.com/ethz-asl/rovio

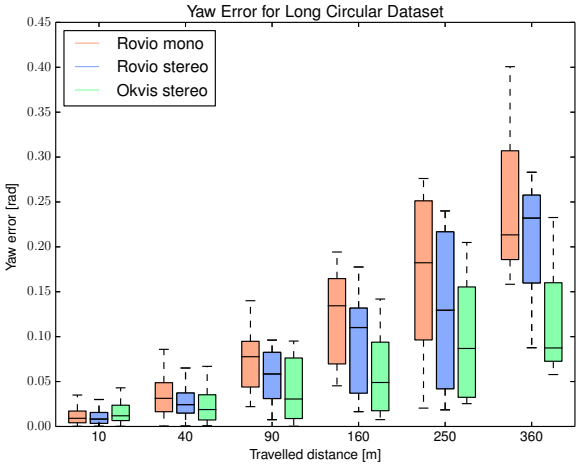

Figure 11.17: Accumulated yaw error over traveled distance for the long circular dataset. Red: baseline monocular Rovio. Blue: baseline stereo Rovio. Green: stereo Okvis with online parameter estimation.

9 Bearing Vector Calculus

Assuming a stationary 3D point landmark $\mathcal{F}$ with bearing vector $\boldsymbol{\mu}$ and distance parameter ρ, the corresponding differential equations can be obtained by totally differentiating the kinematics:

$$\frac{d}{dt}\Big\{{}_{\mathcal{I}}\boldsymbol{r}_{\mathcal{I}\mathcal{F}} = {}_{\mathcal{I}}\boldsymbol{r}_{\mathcal{I}\mathcal{C}} + \boldsymbol{q}_{\mathcal{C}\mathcal{I}}^{-1}(\boldsymbol{\mu}\, d(\rho))\Big\}. \tag{11.75}$$

For this we requires the following partial differentials:

$$\frac{d}{dt}({}_{\mathcal{I}}\boldsymbol{r}_{\mathcal{I}\mathcal{C}}) = {}_{\mathcal{I}}\boldsymbol{v}_{\mathcal{C}}, \tag{11.76}$$

$$\frac{\partial}{\partial \boldsymbol{q}_{\mathcal{C}\mathcal{I}}}(\boldsymbol{q}_{\mathcal{C}\mathcal{I}}^{-1}(\boldsymbol{\mu}\, d(\rho))) = -\boldsymbol{q}_{\mathcal{C}\mathcal{I}}^{-1}(\boldsymbol{\mu}\, d(\rho))^{\times}\boldsymbol{C}(\boldsymbol{q}_{\mathcal{C}\mathcal{I}}^{-1}), \tag{11.77}$$

$$= -\boldsymbol{C}(\boldsymbol{q}_{\mathcal{C}\mathcal{I}}^{-1})\boldsymbol{\mu}^{\times} d(\rho), \tag{11.78}$$

$$\frac{d}{dt}\boldsymbol{q}_{\mathcal{C}\mathcal{I}} = \boldsymbol{\omega}_{\mathcal{C}}, \tag{11.79}$$

$$\frac{\partial}{\partial \boldsymbol{\mu}}(\boldsymbol{q}_{\mathcal{C}\mathcal{I}}^{-1}(\boldsymbol{\mu}\, d(\rho))) = \boldsymbol{C}(\boldsymbol{q}_{\mathcal{C}\mathcal{I}}^{-1})\boldsymbol{\mu}^{\times}\boldsymbol{N}(\boldsymbol{\mu})d(\rho), \tag{11.80}$$

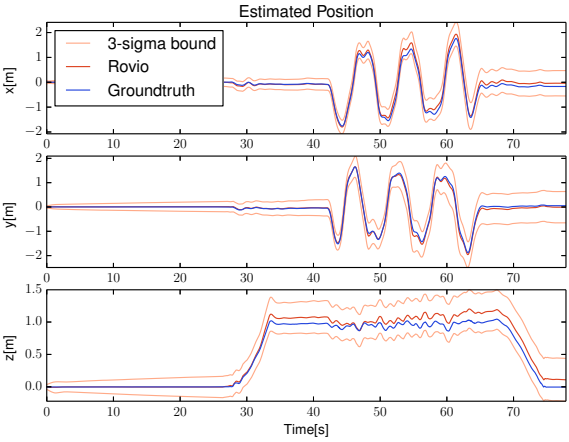

Figure 11.18: Estimated MAV position for aggressive flight. Red: estimated position. Light-red: 3-σ bounds. Blue: Vicon groundtruth. The unobservable position accumulates uncertainty over time.

$$\frac{\partial}{\partial \rho}(\boldsymbol{q}_{\mathcal{CI}}^{-1}(\boldsymbol{\mu}\, d(\rho))) = \boldsymbol{C}(\boldsymbol{q}_{\mathcal{CI}}^{-1})\boldsymbol{\mu} d'(\rho). \tag{11.81}$$

In eq. (11.77) we use the chain rule together with eqs. (11.14) and (11.15), in eq. (11.79) we directly employed eq. (11.13), and eq. (11.80) relies on eq. (11.30). The total differential then yields and can be simplified to (left multiplication with $\boldsymbol{C}(\boldsymbol{q}_{\mathcal{CI}})$):

$$\begin{aligned} 0 &= {}_{\mathcal{I}}\boldsymbol{v}_{\mathcal{C}} - \boldsymbol{C}(\boldsymbol{q}_{\mathcal{CI}}^{-1})\boldsymbol{\mu}^{\times}\, \boldsymbol{\omega}_{\mathcal{C}} d(\rho) \\ &\quad + \boldsymbol{C}(\boldsymbol{q}_{\mathcal{CI}}^{-1})\big(\boldsymbol{\mu}^{\times}\boldsymbol{N}(\boldsymbol{\mu})\dot{\boldsymbol{\mu}}\, d(\rho) + \boldsymbol{\mu}\, d'(\rho)\, \dot{\rho}\big), \end{aligned} \tag{11.82}$$

$$\begin{aligned} 0 &= \boldsymbol{v}_{\mathcal{C}} - \boldsymbol{\mu}^{\times}\boldsymbol{\omega}_{\mathcal{C}}\, d(\rho) \\ &\quad + \boldsymbol{\mu}^{\times}\boldsymbol{N}(\boldsymbol{\mu})\dot{\boldsymbol{\mu}}\, d(\rho) + \boldsymbol{\mu}\, d'(\rho)\, \dot{\rho}. \end{aligned} \tag{11.83}$$

From this the dynamics for the bearing vector and distance parameter can be obtained by pre-multiplication with $1/d(\rho)\boldsymbol{N}(\boldsymbol{\mu})^T\boldsymbol{\mu}^{\times}$ and $1/d'(\rho)\boldsymbol{\mu}^T$ respectively:

$$\dot{\boldsymbol{\mu}} = \boldsymbol{N}(\boldsymbol{\mu})^T\left(\hat{\boldsymbol{\omega}}_{\mathcal{C}} + \boldsymbol{n}(\boldsymbol{\mu})^{\times}\frac{\hat{\boldsymbol{v}}_{\mathcal{C}}}{d(\rho)}\right) + \boldsymbol{w}_{\mu}, \tag{11.84}$$

$$\dot{\rho} = -\boldsymbol{n}(\boldsymbol{\mu})^T\hat{\boldsymbol{v}}_{\mathcal{C}}/d'(\rho) + w_{\rho}. \tag{11.85}$$

Here we used the identities $\boldsymbol{N}(\boldsymbol{\mu})^T\boldsymbol{\mu}^{\times}\boldsymbol{\mu}^{\times} = -\boldsymbol{N}(\boldsymbol{\mu})^T$ and $\boldsymbol{N}(\boldsymbol{\mu})^T\boldsymbol{N}(\boldsymbol{\mu}) = \boldsymbol{I}$. Also, some additive process noise has been added.

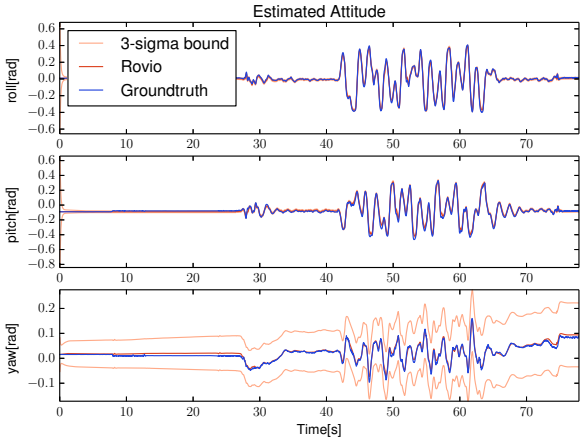

Figure 11.19: Estimated MAV attitude for aggressive flight. The yaw-pitch-roll decomposition is employed to separate the unobservable yaw from the two inclination angles (only visualization). Red: estimated attitude. Light-red: 3-σ bounds. Blue: Vicon groundtruth.

Applying the Euler-forward discretization scheme on the continuous time differential equation of the bearing vectors (11.65) yields:

$$\begin{aligned}
\boldsymbol{\mu}_{k+1} &= \boldsymbol{\mu}_k \boxplus \Delta t\Big(\boldsymbol{N}(\boldsymbol{\mu}_k)^T\Big(\hat{\boldsymbol{\omega}}_{\mathcal{C}} + \boldsymbol{n}(\boldsymbol{\mu}_k)^{\times}\frac{\hat{\boldsymbol{v}}_{\mathcal{C}}}{d(\rho_k)}\Big) + \boldsymbol{w}_{\mu}\Big), \\
\boldsymbol{\mu}_{k+1} &= \exp\Big(\Delta t\boldsymbol{N}(\boldsymbol{\mu}_k)\Big(\boldsymbol{N}(\boldsymbol{\mu}_k)^T\Big(\hat{\boldsymbol{\omega}}_{\mathcal{C}} + \boldsymbol{n}(\boldsymbol{\mu}_k)^{\times}\frac{\hat{\boldsymbol{v}}_{\mathcal{C}}}{d(\rho_k)}\Big) + \boldsymbol{w}_{\mu}\Big)\Big) \otimes \boldsymbol{\mu}_k, \\
\boldsymbol{\mu}_{k+1} &= \exp\Big(\Delta t\Big(\big(\boldsymbol{I} - \boldsymbol{n}(\boldsymbol{\mu}_k)\boldsymbol{n}(\boldsymbol{\mu}_k)^T\big)\,\hat{\boldsymbol{\omega}}_{\mathcal{C}} + \boldsymbol{n}(\boldsymbol{\mu}_k)^{\times}\frac{\hat{\boldsymbol{v}}_{\mathcal{C}}}{d(\rho_k)} + \boldsymbol{N}(\boldsymbol{\mu}_k)\boldsymbol{w}_{\mu}\Big)\Big) \otimes \boldsymbol{\mu}_k.
\end{aligned} \tag{11.86}$$

Here we applied the definition of boxplus (11.23) and used the identity $\boldsymbol{N}(\boldsymbol{\mu})\boldsymbol{N}(\boldsymbol{\mu})^T = \boldsymbol{I} - \boldsymbol{n}(\boldsymbol{\mu})\boldsymbol{n}(\boldsymbol{\mu})^T$. The three components influencing the bearing vector prediction can be observed here: the perpendicular part of the rotational rate, the linear velocity weighted by the inverse distance, and the additive noise.

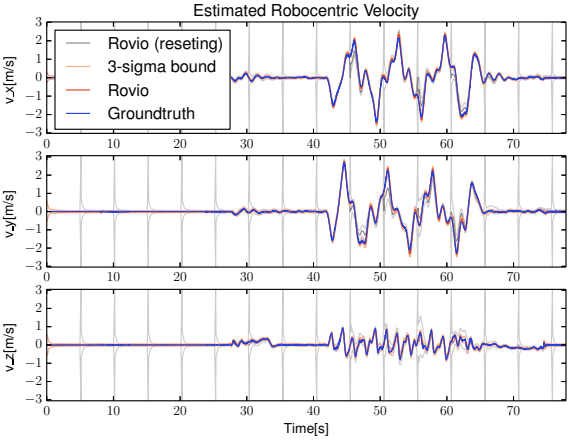

Figure 11.20: Estimated MAV velocity for aggressive flight. Red: estimated velocity. Light-red: 3-σ bounds. Blue: Vicon groundtruth. Grey: estimated velocity when reseting Rovio every 5 s. Since the velocity is expressed in the robocentric IMU coordinate frame it is fully observable and the uncertainty remains bounded.

Bibliography

[1] G. Agamennoni, J. I. Nieto, and E. M. Nebot. An outlier-robust Kalman filter. In *IEEE Int. Conf. on Robotics and Automation*, 2011. doi: 10.1109/ICRA.2011.5979605.

[2] Y. Aoustin, F. Plestan, and V. Lebastard. Experimental comparison of several posture estimation solutions for biped robot Rabbit. In *Robotics and Automation, IEEE International Conference on*, 2008. doi: 10.1109/ROBOT.2008.4543378.

[3] C. Audras, A. I. Comport, M. Meilland, and P. Rives. Real-time dense appearance-based SLAM for RGB-D sensors. In *Australasian Conf. on Robotics and Automation.*, 2011.

[4] T. D. Barfoot. *State Estimation for Robotics: A Matrix Lie Group Approach.* 2016.

[5] B. Bell and F. Cathey. The iterated Kalman filter update as a Gauss-Newton method. *IEEE Transactions on Automatic Control*, 38(2), 1993.

[6] D. Belter and P. Skrzypczynski. Precise self-localization of a walking robot on rough terrain using parallel tracking and mapping. *Industrial Robot-an International Journal*, 40(3), 2013. doi: Doi10.1108/01439911311309924.

[7] F. Bloechliger, M. Bloesch, P. Fankhauser, M. Hutter, and R. Siegwart. Foot-Eye Calibration of Legged Robot Kinematics. In *International Conference on Climbing and Walking Robot*, 2016. doi: 10.3929/ethz-a-010655381.

[8] M. Bloesch and M. Hutter. Technical Implementations of the Sense of Balance. In *Humanoid Robotics: a Reference*, chapter HB. 2017.

[9] M. Bloesch, S. Weiss, D. Scaramuzza, and R. Y. Siegwart. Vision Based MAV Navigation in Unknown and Unstructured Environments. In *IEEE International Conference on Robotics and Automation*, 2010. doi: 10.3929/ethz-a-010137518.

[10] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart. State Estimation for Legged Robots - Consistent Fusion of Leg Kinematics and IMU. In *Robotics Science and Systems Conference*, 2012. doi: 10.15607/RSS.2012.VIII.003.

[11] M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger, and R. Siegwart. State Estimation for Legged Robots on Unstable and Slippery Terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013. doi: 10.1109/IROS.2013.6697236.

[12] M. Bloesch, M. Hutter, C. Gehring, M. A. Hoepflinger, and R. Siegwart. Kinematic Batch Calibration for Legged Robots. In *IEEE International Conference on Robotics and Automation*, 2013. doi: 10.1109/ICRA.2013.6630924.

[13] M. Bloesch, S. Omari, P. Fankhauser, H. Sommer, C. Gehring, J. Hwangbo, M. A. Hoepflinger, M. Hutter, and R. Siegwart. Fusion of Optical Flow and Inertial Measurements for Robust Egomotion Estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014. doi: 10.3929/ethz-a-010184819.

[14] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart. Robust Visual Inertial Odometry Using a Direct EKF-Based Approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015. doi: 10.3929/ethz-a-010566547.

[15] M. Bloesch, S. Omari, and A. Jaeger. ROVIO: RObust Visual-Inertial Odometry, 2015.

[16] M. Bloesch, H. Sommer, T. Laidlow, M. Burri, G. Nützi, P. Fankhauser, D. Bellicoso, C. Gehring, S. Leutenegger, M. Hutter, and R. Siegwart. A Primer on the Differential Calculus of 3D Orientations. *CoRR*, abs/1606.0, 2016.

[17] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart. IEKF-based Visual-Inertial Odometry using Direct Photometric Feedback. *International Journal of Robotics Research*, (conditionally accepted), 2017.

[18] M. Buehler, R. Playter, and M. Raibert. Robots Step Outside. In *International Symposium on Adaptive Motion of Animals and Machines*, 2005.

[19] M. Burri, M. Bloesch, D. Schindler, I. Gilitschenski, Z. Taylor, and R. Siegwart. Generalized Information Filtering for MAV Parameter Estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016. doi: 10.1109/IROS.2016.7759483.

[20] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart. The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 2016. doi: 10.1177/0278364915620033.

[21] J. A. Castellanos, J. Neira, and J. D. Tardos. Limits to the consistency of EKF-based SLAM. In *IFAC Symposium on Intelligent Autonomous Vehicles*, 2004. doi: 10.1109/TAC.2000.880989.

[22] Z. Chen, K. Jiang, and J. C. Hung. Local observability matrix and its application to observability analyses. In *Industrial Electronics Society, 16th Annual Conference of IEEE*, volume 1, 1990. doi: 10.1109/IECON.1990.149118.

[23] A. Chilian, H. Hirschmüller, and M. Görner. Multisensor data fusion for robust pose estimation of a six-legged walking robot. In *IEEE International Conference on Intelligent Robots and Systems*, 2011. doi: 10.1109/IROS.2011.6048125.

[24] S. Chitta, P. Vernaza, R. Geykhman, and D. D. Lee. Proprioceptive localization for a quadrupedal robot on known terrain. In *Robotics and Automation, IEEE International Conference on*, 2007. doi: 10.1109/ROBOT.2007.364185.

[25] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel. 1-point RANSAC for EKF-based Structure from Motion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009. doi: 10.1109/IROS.2009.5354410.

[26] L. Clemente, A. Davison, I. Reid, J. Neira, and J. Tardós. Mapping Large Loops with a Single Hand-Held Camera. In *Proceedings of Robotics: Science and Systems*, 2007.

[27] J. A. Cobano, J. Estremera, and P. Gonzalez de Santos. Location of legged robots in outdoor environments. *Robotics and Autonomous Systems*, 56(9), 2008. doi: 10.1016/j.robot.2007.12.003.

[28] A. J. Davison. Real-Time Simultaneous Localisation and Mapping with a Single Camera. In *IEEE International Conference on Computer Vision*, 2003. doi: 10.1109/ICCV.2003.1238654.

[29] D. D. Diel, P. DeBitetto, and S. Teller. Epipolar Constraints for Vision-Aided Inertial Navigation. In *IEEE Workshops on Application of Computer Vision*, volume 2, 2005. doi: 10.1109/ACVMOT.2005.48.

[30] T. Drummond. Lie groups, Lie algebras, projective geometry and optimization for 3D Geometry, Engineering and Computer Vision, 2014.

[31] E. Eade. Lie groups for Computer Vision, 2014.

[32] N. El-Sheimy, H. Hou, and X. Niu. Analysis and Modeling of Inertial Sensors Using Allan Variance. *IEEE Trans. on Instrumentation and Measurement*, 57 (1), 2008. doi: 10.1109/TIM.2007.908635.

[33] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *European Conference on Computer Vision*, 2014.

[34] J. Englsberger, A. Werner, C. Ott, B. Henze, M. A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, and K. Schmid. Overview of the torque-controlled humanoid robot TORO. In *IEEE International Conference on Humanoid Robots*, 2014. doi: 10.1109/HUMANOIDS.2014.7041473.

[35] M. F. Fallon, M. Antone, N. Roy, and S. Teller. Drift-free humanoid state estimation fusing kinematic, inertial and LIDAR sensing. In *IEEE-RAS International Conference on Humanoid Robots*, 2014. doi: 10.1109/HUMANOIDS.2014.7041346.

[36] P. Fankhauser, M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, and R. Siegwart. Reinforcement Learning of Single Legged Locomotion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013. doi: 10.3929/ethz-a-010018685.

[37] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart. Robot-Centric Elevation Mapping With Uncertainty Estimates. In *International Conference on Climbing and Walking Robot*, 2014. doi: 10.3929/ethz-a-010173654.

[38] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart. Kinect v2 for Mobile Robot Navigation: Evaluation and Modeling. In *International Conference on Advanced Robotics*, 2015. doi: 10.3929/ethz-a-010513824.

[39] P. Fankhauser, M. Bloesch, P. A. Krüsi, R. Diethelm, M. Wermelinger, T. Schneider, M. T. Dymczyk, M. Hutter, and R. Siegwart. Collaborative Navigation for Flying and Walking Robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016. doi: 10.3929/ethz-a-010687710.

[40] C. Forster, M. Pizzoli, and D. Scaramuzza. SVO : Fast Semi-Direct Monocular Visual Odometry. In *IEEE International Conference on Robotics and Automation*, 2014. doi: 10.1109/ICRA.2014.6906584.

[41] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-Manifold Preintegration Theory for Fast and Accurate Visual-Inertial Navigation. 2016.

[42] A. A. Frank. Automatic Control Systems for Legged Locomotion. Technical Report May, 1968.

[43] B. Gaßmann, F. Zacharias, J. M. Zöllner, and R. Dillmann. Localization of walking robots. In *Proceedings - IEEE International Conference on Robotics and Automation*, number April, 2005. doi: 10.1109/ROBOT.2005.1570322.

[44] S. Gehrig, F. Eberli, and T. Meyer. A Real-Time Low-Power Stereo Vision Engine Using Semi-Global Matching. *Computer Vision Systems*, 5815, 2009.

[45] C. Gehring, S. Coros, M. Hutter, M. Bloesch, M. A. Hoepflinger, and R. Siegwart. Control of Dynamic Gaits for a Quadrupedal Robot. In *IEEE International Conference on Robotics and Automation*, 2013. doi: 10.3929/ethz-a-010023052.

[46] C. Gehring, S. Coros, M. Hutter, M. Bloesch, P. Fankhauser, M. A. Hoepflinger, and R. Y. Siegwart. Towards Automatic Discovery of Agile Gaits for Quadrupedal Robots. In *IEEE International Conference on Robotics and Automation*, 2014. doi: 10.3929/ethz-a-010183016.

[47] C. Gehring, C. Bellicoso, S. Coros, M. Bloesch, P. Fankhauser, M. Hutter, and R. Siegwart. Dynamic Trotting on Slopes for Quadrupedal Robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015. doi: 10.3929/ethz-a-010535713.

[48] C. Gehring, S. Coros, M. Hutler, C. Dario Bellicoso, H. Heijnen, R. Diethelm, M. Bloesch, P. Fankhauser, J. Hwangbo, M. Hoepflinger, and R. Siegwart. Practice Makes Perfect: An Optimization-Based Approach to Controlling Agile Motions for a Quadruped Robot. *IEEE Robotics and Automation Magazine*, 23 (1), 2016. doi: 10.1109/MRA.2015.2505910.

[49] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Computer Vision and Pattern Recognition*, 2012.

[50] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(October), 2013. doi: 10.1177/0278364913491297.

[51] M. Görner and A. Stelzer. A leg proprioception based 6 DOF odometry for statically stable walking robots. *Autonomous Robots*, 34(4), 2013. doi: 10.1007/s10514-013-9326-3.

[52] M. Grewal and A. Andrews. *Kalman filtering: theory and practice using MATLAB*, volume 5. 2 edition, 2001.

[53] O. Gür and U. Saranli. Model-based proprioceptive state estimation for spring-mass running. In *Robotics: Science and Systems VII*, 2012.

[54] W. R. Hamilton. On quaternions; or on a new system of imaginaries in algebra. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 25(163), 1844.

[55] A. Handa, M. Bloesch, V. Patraucean, S. Stent, J. McCormac, and A. Davison. Gvnn: Neural Network Library for Geometric Computer Vision. In *European Conference on Computer Vision*, 2016. doi: 10.1007/978-3-319-49409-8_9.

[56] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15. Citeseer, 1988.

[57] R. Hermann and A. Krener. Nonlinear controllability and observability. *IEEE Trans. on Automatic Control*, 22(5), 1977. doi: 10.1109/TAC.1977.1101601.

[58] L. Hertig, D. Schindler, M. Bloesch, C. D. Remy, and R. Siegwart. Unified State Estimation for a Ballbot. In *IEEE International Conference on Robotics and Automation*, 2013. doi: 10.1109/ICRA.2013.6630913.

[59] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1), 2011.

[60] M. A. Hoepflinger, M. Hutter, C. Gehring, M. Bloesch, and R. Siegwart. Unsupervised Identification and Prediction of Foothold Robustness. In *IEEE International Conference on Robotics and Automation*, 2013. doi: 10.1109/ICRA.2013.6631036.

[61] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis. Analysis and improvement of the consistency of extended Kalman filter based SLAM. In *IEEE International Conference on Robotics and Automation*, 2008. doi: 10.1109/ROBOT.2008.4543252.

[62] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis. Observability-based Rules for Designing Consistent EKF SLAM Estimators. *International Journal of Robotics Research*, 29(5), 2010. doi: http://dx.doi.org/10.1177/0278364909353640.

[63] M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, C. D. Remy, and R. Siegwart. StarlETH: A Compliant Quadrupedal Robot for Fast, Efficient, and Versatile Locomotion. In *International Conference on Climbing and Walking Robot*, 2012. doi: 10.3929/ethz-a-010034688.

[64] M. Hutter, M. A. Hoepflinger, C. Gehring, M. Bloesch, C. D. Remy, and R. Siegwart. Hybrid Operational Space Control for Compliant Legged Systems. In *Robotics Science and Systems Conference*, 2012. doi: 10.3929/ethz-a-010184796.

[65] M. Hutter, M. Bloesch, J. Buchli, C. Semini, S. Bazeille, L. Righetti, and J. Bohg. AGILITY – Dynamic Full Body Locomotion and Manipulation with Autonomous Legged Robots. In *IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2013. doi: 10.3929/ethz-a-009996472.

[66] M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, and R. Y. Siegwart. Walking and Running with StarlETH. In *International Symposium on Adaptive Motion of Animals and Machines*, 2013. doi: 10.3929/ethz-a-010022793.

[67] M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, P. Fankhauser, and R. Y. Siegwart. Excitation and Stabilization of Passive Dynamics in Locomotion using Hierarchical Operational Space Control. In *IEEE International Conference on Robotics and Automation*, 2014. doi: 10.3929/ethz-a-010184874.

[68] M. Hutter, C. Gehring, M. A. Hoepflinger, M. Bloesch, and R. Siegwart. Toward Combining Speed, Efficiency, Versatility, and Robustness in an Autonomous Quadruped. *IEEE Transactions on Robotics*, 30(6), 2014. doi: 10.1109/TRO.2014.2360493.

[69] M. Hutter, H. Sommer, C. Gehring, M. A. Hoepflinger, M. Bloesch, and R. Siegwart. Quadrupedal Locomotion using Hierarchical Operational Space Control. *The International Journal of Robotics Research*, 33(8), 2014. doi: 10.3929/ethz-a-010184871.

[70] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger. ANYmal – a Highly Mobile and Dynamic Quadrupedal Robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016. doi: 10.3929/ethz-a-010686165.

[71] H. Jin, P. Favaro, and S. Soatto. A semi-direct approach to structure from motion. In *The Visual Computer*, volume 19. 2003. doi: 10.1007/s00371-003-0202-6.

[72] M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesman, J. Carff, W. Rifenburgh, P. Kaveti, W. Straatman, J. Smith, M. Griffioen, B. Layton, T. de Boer, T. Koolen, P. Neuhaus, and J. Pratt. Team IHMC's Lessons Learned from the DARPA Robotics Challenge Trials. *Journal of Field Robotics*, 2015. doi: 10.1002/rob.

[73] E. S. Jones and S. Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4), 2011. doi: 10.1177/0278364910388963.

[74] S. J. Julier. The scaled unscented transformation. In *Proc. of the American Control Conference*, volume 6, 2002. doi: 10.1109/ACC.2002.1025369.

[75] S. J. Julier and J. K. Uhlmann. A counter example to the theory of simultaneous localization and map building. In *IEEE International Conference on Robotics and Automation*, 2001. doi: 10.1109/ROBOT.2001.933280.

[76] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2), 2012. doi: 10.1177/0278364911430419.

[77] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1), 1960.

[78] K. Kaneko, F. Kanehiro, S. Kajita, M. Morisawa, K. Fujiwara, K. Harada, and H. Hirukawa. Slip observer for walking on a low friction floor. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2005. doi: 10.1109/IROS.2005.1545184.

[79] J. Kelly and G. S. Sukhatme. Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration. *The International Journal of Robotics Research*, 30(1), 2011. doi: 10.1177/0278364910382802.

[80] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for RGB-D cameras. In *IEEE International Conference on Robotics and Automation*, 2013. doi: 10.1109/ICRA.2013.6631104.

[81] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *IEEE and ACM Int. Symposium on Mixed and Augmented Reality*, 2007. doi: 10.1109/ISMAR.2007.4538852.

[82] V. Lebastard, Y. Aoustin, and F. Plestan. Estimation of absolute orientation for a bipedal robot: Experimental results. *IEEE Transactions on Robotics*, 27 (1), 2011. doi: 10.1109/TRO.2010.2094410.

[83] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart. Keyframe-Based Visual-Inertial SLAM using Nonlinear Optimization. In *Proceedings of Robotics: Science and Systems*, 2013.

[84] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3), 2015. doi: 10.1177/0278364914554813.

[85] M. Li and A. I. Mourikis. High-precision, consistent EKF-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6), 2013. doi: 10.1177/0278364913481251.

[86] P. C. Lin, H. Komsuoglu, and D. E. Koditschek. A leg configuration measurement system for full-body pose estimates in a hexapod robot. *IEEE Transactions on Robotics*, 21(3), 2005. doi: 10.1109/TRO.2004.840898.

[87] P. C. Lin, H. Komsuoglu, and D. E. Koditschek. Sensor data fusion for body state estimation in a hexapod robot with dynamical gaits. *IEEE Trans. on Robotics*, 22(5), 2006. doi: 10.1109/TRO.2006.878954.

[88] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 2004.

[89] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *International Joint Conference on Artificial Intelligence*, 1981.

[90] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart. Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization. In *Proceedings of Robotics: Science and Systems*, 2015. doi: 10.15607/RSS.2015.XI.037.

[91] J. Ma, S. Susca, M. Bajracharya, L. Matthies, M. Malchano, and D. Wooden. Robust multi-sensor, day/night 6-DOF pose estimation for a dynamic legged vehicle in GPS-denied environments. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2012. doi: 10.1109/ICRA.2012.6225132.

[92] J. Ma, M. Bajracharya, S. Susca, L. Matthies, and M. Malchano. Real-time pose estimation of a dynamic quadruped in GPS-denied environments for 24-hour operation. *The International Journal of Robotics Research*, 2015. doi: 10.1177/0278364915587333.

[93] R. Mahony, T. Hamel, and J. M. Pflimlin. Complementary filter design on the special orthogonal group SO(3). *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, CDC-ECC '05*, 2005(1), 2005. doi: 10.1109/CDC.2005.1582367.

[94] R. B. Mcghee. Finite state control of quadruped locomotion. *Simulation*, 1967.

[95] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. RSLAM: A System for Large-Scale Mapping in Constant-Time Using Stereo. *Int. Journal of Computer Vision*, 94(2), 2011. doi: 10.1007/s11263-010-0361-7.

[96] F. M. Mirzaei and S. I. Roumeliotis. A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. *IEEE Trans. on Robotics*, 24(5), 2008.

[97] N. Molton, A. Davison, and I. Reid. Locally Planar Patch Features for Real-Time Structure from Motion. In *British Machine Vision Conference*, 2004.

[98] J. Montiel, J. Civera, and A. Davison. Unified Inverse Depth Parametrization for Monocular SLAM. In *Proceedings of Robotics: Science and Systems*, 2006.

[99] A. I. Mourikis and S. I. Roumeliotis. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. In *IEEE International Conference on Robotics and Automation*, 2007. doi: 10.1109/ROBOT.2007.364024.

[100] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. ORB-SLAM : a Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31 (5), 2015. doi: 10.1109/TRO.2015.2463671.

[101] M. P. Murphy, A. Saunders, C. Moreira, a. a. Rizzi, and M. Raibert. The LittleDog robot. *The International Journal of Robotics Research*, 30(2), 2010. doi: 10.1177/0278364910387457.

[102] M. Neunert, M. Bloesch, and J. Buchli. An Open Source, Fiducial Based, Visual-Inertial Motion Capture System. *International Conference on Information Fusion*, 2015.

[103] M. Neunert, M. Giftthaler, M. Frigerio, C. Semini, and J. Buchli. Fast Derivatives of Rigid Body Dynamics for Control, Optimization and Estimation. In *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, 2016.

[104] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgale, and R. Siegwart. A Synchronized Visual-Inertial Sensor System with FPGA Pre-Processing for Accurate Real-Time SLAM. In *IEEE Int. Conf. on Robotics and Automation*, 2014.

[105] J. Nikolic, P. Furgale, A. Melzer, and R. Siegwart. Maximum Likelihood Identification of Inertial Sensor Noise Model Parameters. *IEEE Sensors Journal*, 16 (1), 2016. doi: 10.1109/JSEN.2015.2476668.

[106] N. Okita and H. J. Sommer. A novel foot slip detection algorithm using unscented Kalman Filter innovation. In *American Control Conference (ACC), 2012*, 2012.

[107] S. Omari and G. Ducard. Metric visual-inertial navigation system using single optical flow feature. In *European Control Conference*, 2013.

[108] S. Omari, M. Bloesch, P. Gohl, and R. Siegwart. Dense Visual-Inertial Navigation System for Mobile Robots. In *IEEE International Conference on Robotics and Automation*, 2015. doi: 10.1109/ICRA.2015.7139554.

[109] R. J. Peterka. Sensorimotor integration in human postural control. *Journal of neurophysiology*, 88(3), 2002. doi: 10.1152/jn.00605.2001.

[110] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34(3), 2013.

[111] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter. BigDog , the Rough-Terrain Quaduped Robot. In *17th IFAC World Congress*, 2008.

[112] M. Reinstein and M. Hoffmann. System Aided By a Legged Odometer. In *International Conference on Robotics and Automation*, 2011. doi: 10.1109/ICRA.2011.5979609.

[113] M. Reinstein and M. Hoffmann. Dead reckoning in a dynamic quadruped robot based on multimodal proprioceptive sensory information. *IEEE Transactions on Robotics*, 29(2), 2012. doi: 10.1109/TRO.2012.2228309.

[114] C. D. Remy, M. Hutter, M. A. Hoepflinger, M. Bloesch, C. Gehring, and R. Siegwart. Quadrupedal Robots with Stiff and Compliant Actuation. *Automatisierungstechnik*, 60(11), 2012. doi: 10.3929/ethz-a-010000217.

[115] E. Rosten and T. Drummond. Machine Learning for High Speed Corner Detection. *Computer Vision – ECCV 2006*, 1, 2006. doi: 10.1007/11744023_34.

[116] G. P. Roston and E. P. Krotkov. Dead Reckoning Navigation For Walking Robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1992. doi: 10.1109/IROS.1992.587401.

[117] N. Rotella, M. Bloesch, L. Righetti, and S. Schaal. State Estimation for a Humanoid Robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014. doi: 10.1109/IROS.2014.6942674.

[118] C. Semini, J. Buchli, M. Frigerio, T. Boaventura, M. Focchi, E. Guglielmino, F. Cannella, N. G. Tsagarakis, and D. G. Caldwell. {HyQ} – A Dynamic Locomotion Research Platform. In *Proc. of Int. Workshop on Bio-Inspired Robots, Nantes (France)*, 2011.

[119] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar. Initialization-free monocular visual-inertial estimation with application to autonomous MAVs. In *International Symposium on Experimental Robotics*, 2014.

[120] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition*, 1994. doi: 10.1109/CVPR.1994.323794.

[121] B. Siciliano and O. Khatib. Springer handbook of robotics, 2008.

[122] G. Silveira, E. Malis, and P. Rives. An Efficient Direct Approach to Visual SLAM. *IEEE Transactions on Robotics*, 24(5), 2008. doi: 10.1109/TRO.2008.2004829.

[123] S. Singh, P. Csonka, and K. Waldron. Optical Flow Aided Motion Estimation for Legged Locomotion. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006. doi: 10.1109/IROS.2006.282134.

[124] S. P. N. Singh and K. J. Waldron. Attitude Estimation for Dynamic Legged Locomotion Using Range and Inertial Sensors. In *Robotics and Automation, IEEE International Conference on*, 2005. doi: 10.1109/ROBOT.2005.1570352.

[125] J. Sola. Quaternion kinematics for the error-state KF. *Laboratoire d'Analyse et d'Architecture des Systemes-Centre national de la recherche scientifique (LAAS-CNRS), Toulouse, France, Tech. Rep*, 2012.

[126] J. Solà, T. Vidal-Calleja, J. Civera, and J. M. M. Montiel. Impact of landmark parametrization on monocular EKF-SLAM with points and lines. *International Journal of Computer Vision*, 97, 2012. doi: 10.1007/s11263-011-0492-5.

[127] H. W. Sorenson and D. L. Alspach. Recursive bayesian estimation using gaussian sums. *Automatica*, 7(4), 1971. doi: 10.1016/0005-1098(71)90097-5.

[128] A. Stelzer, H. Hirschmuller, and M. Gorner. Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain. *The International Journal of Robotics Research*, 31(4), 2012. doi: 10.1177/0278364911435161.

[129] B. J. Stephens. State estimation for force-controlled humanoid balance using simple models in the presence of modeling error. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2011. doi: 10.1109/ICRA.2011.5980358.

[130] H. Strasdat, J. M. M. Montiel, and A. J. Davison. Real-time monocular SLAM: Why filter? In *IEEE International Conference on Robotics and Automation*, 2010. doi: 10.1109/ROBOT.2010.5509636.

[131] H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige. Double window optimisation for constant time visual SLAM. In *IEEE Int. Conference on Computer Vision*, 2011. doi: 10.1109/ICCV.2011.6126517.

[132] P. Tanskanen, T. Naegeli, M. Pollefeys, and O. Hilliges. Semi-Direct EKF-based Monocular Visual-Inertial Odometry. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

[133] J.-A. Ting, E. Theodorou, and S. Schaal. A Kalman filter for robust outlier detection. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007. doi: 10.1109/IROS.2007.4399158.

[134] F. L. Tong and M. Q. H. Meng. Localization for legged robot with single low-resolution camera using genetic algorithm. In *IEEE ICIT 2007 - 2007 IEEE International Conference on Integration Technology*, 2007. doi: 10.1109/ICITECHNOLOGY.2007.4290452.

[135] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle Adjustment — A Modern Synthesis. In *Vision Algorithms: Theory and Practice*, chapter Lecture No. 2000. doi: 10.1007/3-540-44480-7_21.

[136] V. Usenko, J. Engel, J. Stückler, and D. Cremers. Direct Visual-Inertial Odometry with Stereo Cameras. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2016.

[137] C. Van Loan. Computing integrals involving the matrix exponential. *Automatic Control, IEEE Transactions on*, 23(3), 1978. doi: 10.1109/TAC.1978.1101743.

[138] L. Wagner, P. Fankhauser, M. Bloesch, and M. Hutter. Foot Contact Estimation for Legged Robots in Rough Terrain. In *International Conference on Climbing and Walking Robot*, 2016. doi: 10.3929/ethz-a-010643823.

[139] S. Weiss, M. Achtelik, S. Lynen, L. Kneip, M. Chli, and R. Siegwart. Monocular Vision for Long-term Micro Aerial Vehicle State Estimation: A Compendium. *Journal of Field Robotics*, 30(5), 2013.

[140] M. West. Robust sequential approximate Bayesian estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 43(2), 1981.

[141] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. a. Rizzi, and M. Raibert. Autonomous navigation for BigDog. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2010. doi: 10.1109/ROBOT.2010.5509226.

[142] Xinjilefu and C. G. Atkeson. State estimation of a walking humanoid robot. In *IEEE International Conference on Intelligent Robots and Systems*, 2012. doi: 10.1109/IROS.2012.6386070.

[143] X. Xinjilefu, S. Feng, and C. G. Atkeson. Dynamic State Estimation using Quadratic Programming. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014. doi: 10.1109/IROS.2014.6942679.

[144] X. Xinjilefu, S. Feng, W. Huang, and C. G. Atkeson. Decoupled state estimation for humanoids using full-body dynamics. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014. doi: 10.1109/ICRA.2014.6906609.

Curriculum Vitae

Michael Andre Bloesch
born August 4, 1987
citizen of Moerigen BE, Switzerland

2012–2016	*ETH Zurich, Switzerland* Doctoral studies at the Autonomous Systems Lab; Supervised by Prof. Roland Siegwart, Prof. Andrew Davison and Prof. Marco Hutter
2012–2015	*University of Zurich, Switzerland* Bachelor of Science in Medicine (summa cum laude)
2009–2011	*ETH Zurich, Switzerland* Master of Science in Mechanical Engineering (with distinction)
2005–2009	*ETH Zurich, Switzerland* Bachelor of Science in Mechanical Engineering
2002–2005	*Gymnasium Köniz, Switzerland* Matura in Physics and Applied Mathematics
1993–2002	*Ecole cantonale de langue française à Bern, Switzerland*