POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

State Estimation for Legged Robots: Fusion of Inertial Sensing and Model-Based Control Information

Tesi di Laurea Magistrale in
Automation and Control Engineering - Ingegneria dell'Automazione

Author: **Stefano Taborelli**

Student ID: 259076
Advisor: Prof. Marco Faroni
Co-advisors: Chris Mcgreavy
Academic Year: 2024-25

Abstract

Accurate state estimation is a fundamental requirement for stable locomotion in quadruped robots governed by model predictive control architectures. Purely inertial pose estimation based on inertial measurement unit (IMU) measurements inevitably suffers from drift. To mitigate this effect, most approaches rely on exteroceptive sensors or contact-based constraints from leg kinematics, increasing system complexity.

This thesis investigates an alternative strategy: exploiting robot controller-generated quantities as an additional information source within the state estimator. The central research question is whether outputs from a nonlinear model predictive controller (NMPC) can provide sufficiently informative and stable constraints to support inertial data in an Extended Kalman Filter (EKF) framework.

The work begins with the experimental characterization of a real IMU, focusing on noise and bias analysis, followed by the development of an IMU-only estimation framework. Building on this foundation, two fusion architectures combining IMU data and NMPC outputs are designed and evaluated in simulation for quadruped pose estimation.

The results suggests that corrections based on controller-derived quantities introduce structural limitations and instability when heavily weighted in the estimation framework. In both tested approaches, reliable performance is achieved only when inertial information dominates, highlighting the limited ability of controller outputs to independently constrain drift. The analysis also emphasizes the importance of using direct, independent measurements and explicit bias modeling for more robust and long-term estimation.

Overall, the study clarifies the limitations of controller-based corrections and provides guidelines for future developments toward contact-aided estimation architecture suitable for real-world quadruped locomotion.

Keywords: Quadruped robots, State estimation, Inertial Measurement Navigation, Extended Kalman Filter, Model Predictive Control

Abstract in lingua italiana

Nei robot quadrupedi controllati tramite Model Predictive Control, la qualità della locomozione dipende in modo critico dall'accuratezza della stima dello stato del robot. La stima di posizione e orientamento ottenuta esclusivamente da misure inerziali, fornite da accelerometri e giroscopi, è soggetta a deriva nel tempo. Per contrastare questo problema, le soluzioni più diffuse in letteratura prevedono l'integrazione di sensori esterni oppure l'impiego di vincoli di contatto e tecniche di odometria basate sulla cinematica dei vari giunti, con un conseguente aumento della complessità del sistema.
Questa tesi esplora una strategia alternativa cercando di sfruttare le grandezze generate dal controllore del robot come fonte informativa aggiuntiva all'interno dello stimatore. La domanda centrale che questa tesi cerca di chiarire è se le uscite del controllore del robot possano fornire vincoli sufficientemente stabili da supportare i dati inerziali in un framework basato su Extended Kalman Filter (EKF).
Il lavoro inizia con la caratterizzazione sperimentale di una IMU reale, seguita dallo sviluppo di un'architettura di stima basata esclusivamente su IMU. Su queste basi vengono quindi progettate e valutate in simulazione due architetture per la stima della posa di un robot quadrupede, che combinano le misure inerziali con le uscite del controllore.
I risultati mostrano che le correzioni basate su quantità derivate dal controllore introducono fenomeni di instabilità quando vengono pesate in modo significativo nel processo di stima, fino a generare la divergenza della stima stessa. In entrambe le configurazioni analizzate, hanno evidenziato la limitata capacità delle uscite del controllore di vincolare autonomamente gli stati. L'analisi ha sottolineato inoltre l'importanza di utilizzare misure dirette e indipendenti e di modellizzare esplicitamente i bias per ottenere una stima più robusta e consistente nel lungo termine.
Nel complesso, lo studio mette in luce i limiti dell'impiego dell'output del controllore nel processo di stima e fornisce indicazioni per sviluppi futuri orientati verso un'architettura di stima più affermata, che sfrutti i contatti e l'odometria basata sulla cinematica dei vari giunti.

Keywords: Robot quadrupedi, Stima dello stato, Navigazione inerziale, Extended Kalman Filter, Model Predictive Control

Contents

1 | Introduction

Robotics plays an increasingly important role in industrial environments where human intervention is either unsafe, impractical, or inefficient. This is particularly true in large-scale scientific infrastructures such as CERN (European Organization For Nuclear Research), where robotic systems are routinely employed for inspection, maintenance, and intervention tasks in environments characterized by radiation hazards, confined spaces, and restricted accessibility. Within this context, quadruped robots represent a promising solution for use cases where uneven terrain, stairs or other obstacles are present.

At CERN, almost all robotic platforms, including quadruped robots, are developed within the CERN Robotic Framework (CRF), a modular in-house architecture tailored to the facility's operational constraints. In the case of quadrupeds, locomotion is governed by a Nonlinear Model Predictive Controller (NMPC) and a Whole-Body Controller (WBC), still in development inside the CRF. This controller depends on the availability of an accurate and consistent state estimate at every control cycle, and for this reason state estimation (particularly position and orientation) is not only a navigation problem, but a fundamental prerequisite for stable locomotion and it is essential for control.

Motivation and Objective

Quadruped robots at CERN operate in indoor and underground environments where GNSS is unavailable and electromagnetic disturbances are present. In this context, Inertial Measurement Units (IMUs) constitute the primary building block for pose estimation, however, it cannot be used alone since it is affected by drift due to noise and bias accumulation.

State-of-the-art legged robot estimators typically mitigate this issue by fusing IMU data with contact constraints, leg kinematics, or exteroceptive sensing within an EKF-based (extended Kalman filter) framework. Although effective, these approaches increase modeling and implementation complexity.

This thesis investigates the alternative of exploiting motion quantities generated by the NMPC controller itself as an additional information source for state estimation and

whether they can provide sufficiently stable constraints to support inertial state estimation.

To address this question, the work progressively builds from basic inertial estimation, to some IMU-only correction techniques moving toward increasingly integrated architectures combining IMU data and NMPC outputs.

The purpose is not to deliver a finalized estimator, but rather to critically assess the conceptual validity and structural limitations of exploiting controller outputs within a probabilistic state estimation framework.

Structure of the thesis

The thesis is structured into the following chapters.

Chapter 2 first provides an overview of the CERN section in which the work has been developed, then analyzes the state of the art reviewing inertial navigation principles, and discusses state estimation techniques for mobile and legged robots.

Chapter 3 presents the methodology. It describes the hardware setup and the experimental characterization of the IMU. Then it explains the formulation of the IMU-only EKF with ZUPT corrections, and the development of the two fusion strategies combining IMU and NMPC data.

Chapter 4 reports and discusses the experimental and simulation results. It first analyzes signal pre-processing techniques, then evaluates the IMU-only estimator against vision-based ground truth, and finally compares the two proposed fusion architectures in simulation.

Chapter 5 concludes the thesis by summarizing the main findings, highlighting the structural limitations observed in controller-based corrections, and outlining future developments toward more robust contact-based estimation frameworks suitable for real-world quadruped locomotion.

2 | State of the art and context

This thesis was developed at the Controls Electronics and Mechatronics (CEM) group at CERN, particularly in the Mechatronics and Robotics (MRO) section. The CEM group develops and maintains a centralized competence in controls hardware and low-level software as well as focusing on electronics design and production, radiation tolerance, mechatronics and robotics, test and measurement systems. It is responsible for multiple tasks including:

- Design, installation, control and maintenance of the mechatronics control systems for Beam Intercepting Devices in the CERN accelerator complex and experimental facilities (BIDs are components placed in a particle beamline to intentionally intercept, absorb, or shape the beam for machine protection, cleaning, or disposal purposes).
- Development and support of industrial automation solutions for mechatronics based on PLCs, standard off the shelf as well as ad hoc in-house developed actuators/sensors.
- Definition of intervention procedures both for remote maintenance and human intervention procedures coordination.
- Development of novel robotic solutions for remote robotic inspection and telemanipulation in hazardous environments including Artificial Intelligence R&D activity.

In many of those tasks some challenges, specific of the CERN operational environment have to be faced, like robust operation with radiation and electromagnetic interference, control of actuators over long cables and inaccessible environment (size constraints, or dangerous environments). For some of these cases robotics can have a key importance to increase safety and machine availability, perform repetitive, unplanned and dangerous tasks. It is precisely in this context that the MRO section works to implement and maintain robotics and mechatronics solution.
One important use of robotics at CERN is to avoid and prevent the exposure of humans and personnel to hazard. One of the main hazards in CERN facilities is exposure to

ionizing radiation, and this is particularly relevant in particles accelerator underground tunnels, where high-energy particle beams interact with materials in the beamline and surrounding infrastructure, producing induced radioactivity and generating residual radiation even after the beams are switched off. Radiation hazards are not limited to the tunnels: activated components, experimental areas, service caverns, and certain technical zones can also exhibit high radiation levels depending on machine operation and past beam conditions. For this reason, access to all these areas is strictly controlled and often limited in time. A key advantage of using robotic systems in this context is the significant reduction of human exposure to radiation. Robots can perform inspections, measurements, and interventions in activated environments, allowing personnel to carry out their work safely from outside restricted zones.

More specific robotic technologies and examples of their usage at CERN are [8]:

Remote inspections, tele-operation and maintenance
Robots allow intervention during long shutdowns and technical stops. They are used to inspect and verify the conditions of the tunnel infrastructure over the years, enabling to find possible problems and defects. The possibility to remotely manipulate objects enables to perform maintainance in remote or dangerous areas. Those operations are mainly performed using the CERNBOT (Figure 2.2), the MIRA robot (Figure 2.3), the train inspection monorail (TIM) (Figure 2.1) and recently also using quadrupeds to reach areas that are difficult to access (Figure 2.4).

Quadrupeds are different from wheeled robots because, thanks to their legged locomotion, they are capable of traversing uneven terrain, climbing stairs, stepping over obstacles, and operating in environments where flat surfaces cannot be guaranteed, still maintaining stability. These characteristics make quadruped robots particularly suitable for inspection and maintenance tasks in complex industrial environments where structural degradation, obstacles, or temporary installations may alter the environment.

Figure 2.1: TIM robot: autonomous robotic system operating on a ceiling-mounted monorail in the LHC tunnel (27 km long Large hadron collider), composed by different wagons (modular), enabling remote inspections, radiation monitoring, and intervention during beam stops and shutdowns. (Retrieved from [8]).

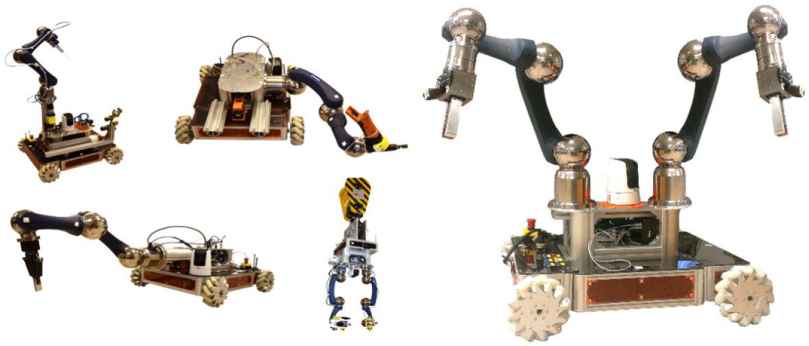

Figure 2.2: CERNbot robot: modular and highly customizable robotic platform for complex interventions in harsh environments. Many robots using the same modular base are currently in use. (Retrieved from [8]).

Figure 2.3: MIRA robot: a compact autonomous mobile robot used for radiation surveying, inspection, and remote operations in the SPS (Super Proton synchrotron) tunnel. (Retrieved from [8]).

Figure 2.4: Quadruped robot used for visual inspection and interventions in remote and inaccessible areas. (Retrieved from [8]).

Quality assurance

Another robotic system developed at CERN is the Automated Robotic Inspection System (ARIS) (Figure 2.5) aiming at automating the visual inspection of the inner surfaces of radiofrequency (RF) accelerating cavities used in CERN's accelerators. These cavities are critical components in which surface defects can degrade performance or cause failure. The automated robotic inspection system is equipped with a high-resolution camera and operates under control of the CERN Robotic Framework, incorporating safety features such as anti-collision mechanisms.

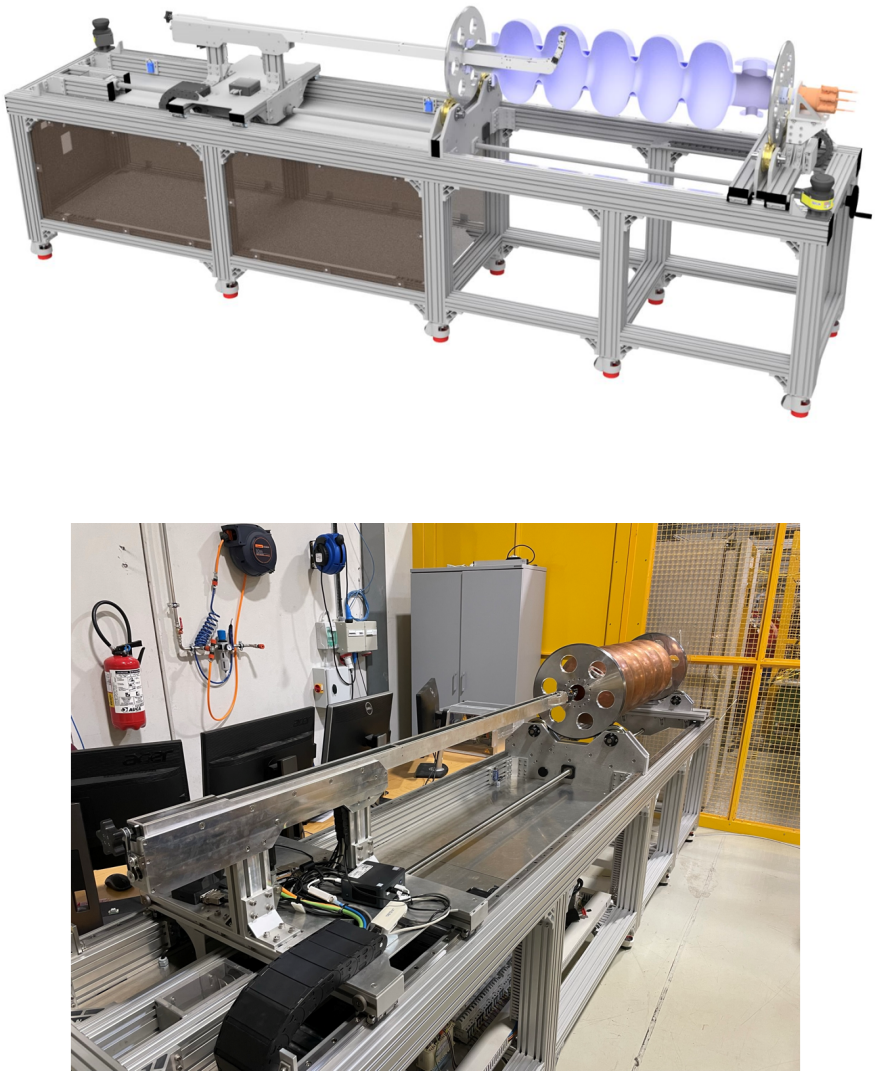

Figure 2.5: Automated Robotic Inspection System (ARIS) while inspecting an RF cavity. (Retrieved from [8]).

Reconnaissance, search and rescue

Robotics plays a very important role in emergency interventions, such as fire or accidents in the tunnels, by performing fire inspection, victim search and initial fire suppression. Some project aims at remotely detect people and their vital signals (MARCHESE project) in disaster scenarios and guide CERN Fire Brigade in their interventions.

Post-mortem analysis of radioactive devices to investigate the causes of failure of components operated in radioactive environments, without exposing personnel to radiation.

2.1. CERN Robotic Framework (CRF)

Almost all the robots used at CERN (apart from industrial robots) use an in-house robotic control system, developed totally at CERN and called CERN Robotic Framework (CRF) [9]. The choice of developing a robotic control system from scratch is guided by the need of modularity and by the need of being able to customize every detail of the software side, not having to be subject to the restriction of third-party closed source software. Industrial robot, for example, have complicated human-robot interfaces demanding operators training, controls are not open to be integrated and customized, communication channel is often via radio signal; details that does not fit the highly niche tasks performed at CERN and the necessity of having user friendly interfaces.
The main goal of this framework is to be able to customize and adapt to every different robot and unusual application, often needed at CERN. It is developed in C++ and structured with many different modules: sensor drivers, middleware and communication, math and utilities modules, motion controllers, navigation and much more.

This framework has been the primary environment in which this thesis was developed, particularly for the sensors-IMU module and the Model Predictive Control one but also exploiting other modules and utilities.

2.2. Inertial navigation, a known challenge

Estimating the position and orientation of a moving system in space is a well known and long-standing problem encountered across a wide range of application domains, from aerospace navigation and pose estimation of aircraft, to mobile robotics and autonomous systems. Depending on the operating environment and the available infrastructure, this problem can be addressed using different sensing and positioning technologies, such as satellite-based navigation systems [1, 15], radio frequency wireless methods such as ultra wideband (UWB) [12] or Radio Frequency Identification (RFID) [26], vision-based methods [6, 25], or inertial sensing [31] ; or most likely combining some of them together.

Among these solutions, inertial navigation plays a central role in scenarios where external references are unavailable, unreliable, or impractical. The most common devices used to address the inertial navigation problem are Inertial Measurement Units (IMUs), which are typically composed of accelerometers and gyroscopes. IMUs provide self-contained motion information that can be exploited to infer the system state, making them particularly suitable for indoor environments, industrial settings, underground facilities or mines.

The fundamental principle of inertial navigation using IMUs is conceptually simple and

can be visualized in Figure 2.6. Linear acceleration measurements are integrated over time to obtain velocity, and velocity is further integrated to estimate position. Similarly, angular rate measurements are integrated to track the system orientation.

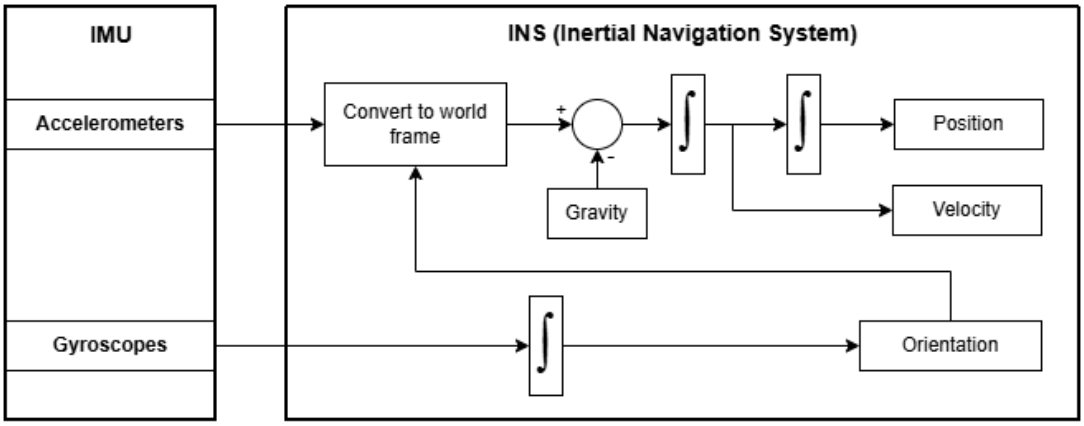

Figure 2.6: Fundamental principle of inertial navigation systems. (Adapted from [24])

However, despite this apparent simplicity, inertial navigation is notoriously challenging in practice. Sensor noise, biases, and modelling inaccuracies are continuously integrated over time, leading to rapid error accumulation and unbounded drift in the estimated states [33].

For this reason, IMUs are rarely used alone in a so known dead-reckoning configuration but instead, they are typically combined with additional information sources or system constraints in order to limit drift and improve estimation accuracy. This fusion can involve external sensors, such as lidars, cameras and GPS, or can exploit knowledge of the system dynamics and operating conditions.

A widely adopted strategy to address the inherent limitations of inertial navigation and fuse multiple sources of information is the use of probabilistic state estimation techniques such as Kalman filtering. By explicitly modelling the system, sensor errors, and associated uncertainties, these methods enable the consistent fusion of system dynamics and available measurements. Kalman filtering and its nonlinear variants provide a framework to propagate uncertainty, correct estimates when new information becomes available, and fuse different sources of information managing the trade-off between them. For these reasons it has become a standard tool in inertial navigation, as can be seen from the numerous related articles and books:[2, 10, 13, 23, 29, 31].
Strategies based on Kalman filtering have also been extensively investigated for dead-reckoning inertial navigation using only IMU measurements. Such approaches have been applied in a variety of application domains, particularly in indoor or underground envi-

ronments where external positioning infrastructure is unavailable or unreliable. One of the most widely studied examples is pedestrian tracking based on shoe-mounted or body-mounted IMUs, where inertial measurements are combined with motion constraints and probabilistic filtering to estimate the user trajectory [10].

A fundamental characteristic of dead-reckoning systems is that estimation errors inevitably accumulate over time. To mitigate these effects, several strategies have been proposed in the literature, including the introduction of motion constraints acting as pseudo-measurements (such as zero-velocity updates), the exploitation of known motion patterns (e.g. walking or running) [17, 32], and the use of probabilistic estimators to explicitly model sensor errors and uncertainties.

Despite these techniques, it is well established that inertial-only navigation remains fundamentally limited in terms of long-term robustness, even with sophisticated filtering and error modeling. The strategies outlined above tend to be effective only in specific application scenarios, such as pedestrian tracking, where the motion exhibits repetitive patterns and frequent stationary phases (for example between 2 steps when one foot can be assumed as stationary) making them not applicable in a broader range of cases. For this reason, dead-reckoning inertial navigation is rarely used as a standalone solution in robotics applications.

Nevertheless, the approaches developed in the context of inertial dead-reckoning represent an important foundation for more advanced state estimation systems. In this work, this approach constitutes the starting point for understanding the behavior and limitations of inertial measurements, and for developing an initial, simplified estimation framework. This preliminary step allows building confidence with the estimation problem before addressing the more general case in which inertial navigation is integrated within a broader state estimation framework and combined with additional information sources to achieve improved robustness and consistency.

2.3. State estimation for mobile robots

State estimation for mobile robots has been extensively studied in the literature, particularly in the context of wheeled platforms operating on structured or semi-structured environments. In these systems, the estimation of the robot pose is commonly supported by wheel odometry, which provides a direct measurement of the robot displacement by integrating wheel encoder readings over time [5, 30].

Wheel odometry offers a strong source of relative motion information, especially at low

speeds and on flat terrain, and is often combined with inertial measurements to improve short-term accuracy and robustness. Typical state estimation pipelines for wheeled robots fuse wheel encoder data, IMU measurements, and exteroceptive sensors such as cameras or lidars using probabilistic frameworks, most commonly based on Extended Kalman Filters, error-state filters, or optimization-based approaches [13, 23].

Wheel odometry is affected by some limitations, including wheel slip, uneven terrain, and unmodeled contact dynamics, which lead to drift over time. For this reason, wheeled robot localization systems that aim at long term high accuracy usually rely also on external references, such as landmarks, maps, or GPS when available, to bound the accumulation of error. Even considering these small limitations, the availability of wheel odometry significantly simplifies the state estimation problem for wheeled robots, as it provides a continuous and directly observable constraint on the robot motion that can be readily incorporated into the estimation framework.

2.4. State estimation for legged robots

State estimation for legged robots presents additional challenges compared to wheeled platforms, as they lack continuous wheel odometry and rely on intermittent and changing contacts with the environment. The support polygon evolves over time, the terrain can be much more unpredictable and uneven, and the robot is subject to impacts and high-frequency dynamics.

As a consequence, the estimation of the base pose and velocity cannot rely on simple kinematic integration of actuator measurements alone. Instead, state estimation for legged robots typically requires the fusion of inertial measurements with kinematic information derived from joint encoders and contact sensors, while explicitly accounting for the contact state of each limb.

A common approach in the literature is to exploit contact constraints during stance phases, assuming that the feet in contact with the ground have zero velocity relative to the environment, fusing this information with the leg odometry [3, 14]. Another explored approach, when reliable contact sensors are not available, consists in estimating the contact state probabilistically. In this framework, contact events are inferred from inertial measurements and incorporated into the estimator as stochastic variables, rather than treated as binary deterministic signals, as proposed in [7]. These constraints can be introduced as pseudo-measurements within a Kalman filter framework, significantly improving observability and reducing drift.

Also in this case, similarly to wheeled robots, several works propose Extended Kalman Filter or error-state formulations that fuse IMU data with leg kinematics and contact information to estimate the robot base state, center of mass motion, and inertial sensor biases. These approaches have been successfully applied to quadruped robots operating in challenging environments, demonstrating improved robustness compared to inertial-only or purely kinematic estimation methods.

Nevertheless, state estimation for legged robots remains a challenging problem, particularly in scenarios involving uneven terrain, slipping contacts, or fast dynamic motions [4]. In this context, the present work builds upon these state-of-the-art techniques and explores different approaches mainly to try integrating the robot controller in the estimation framework.

2.5. Motivations behind the considered study

The application scenario considered in this thesis is related to mobile robots and, more specifically, to legged robots, employed at CERN for inspection, maintenance, and intervention tasks in environments that are unsafe or difficult for human access. In this context, the problem is more appropriately framed as a state estimation problem rather than pure inertial navigation, with the main objective of estimating the state of the robot over time, with particular emphasis on the position and orientation of the robot center of mass. Inertial navigation represents here a fundamental building block, which, together with additional sources of information, enables to build a reliable state estimator.

Accurate state estimation is a key requirement for several aspects of mobile robot operation. To understand why, it is important to know that the utilized quadruped robot is controlled through a Model Predictive Control (MPC) framework, also developed at CERN based on [28]. The MPC controller exploits a dynamic model of the robot, information from onboard sensors (like contact sensors and joint encoders), together with an estimate of the current system state, to predict the future evolution of the robot and compute control inputs for the next time steps to follow a desired reference trajectory [22]. Therefore, it is essential for the control of the robot itself, whether in autonomous or tele-operated modes, to always have an estimate of the current state at every timestep.

State estimation is also critical for precise robot positioning during inspection or manipulation tasks, where the robot must be accurately placed relative to points of interest in the environment. Furthermore, when combined with additional sensing modalities, a reliable state estimate supports mapping and localization tasks (SLAM), enabling the construction of spatial representations of the environment that can be reused for future

operations.

The robot is expected to operate primarily in indoor and underground environments, such as accelerator tunnels and technical galleries, where GNSS is unavailable and strong electromagnetic disturbances are often present, making magnetometer-based heading estimation unreliable or impractical. These constraints motivate the development of a state estimator that relies on inertial sensing and model-based information, remaining robust in GNSS-denied and magnetically disturbed environments.

3 | Methodology

3.1. Objectives

This thesis focuses on state estimation for a quadruped robot operating in challenging environments, for inspection and intervention tasks within CERN facilities. Such environments are typically indoor, underground (e.g., tunnels, technical galleries, and confined industrial areas), where reliable external positioning systems such as GNSS are unavailable. Additionally, these scenarios may present strong electromagnetic disturbances, making the use of magnetometer-based heading estimation unreliable or impractical.

The primary objective of this work is to estimate the pose (position and orientation) of the robot base frame over time. Accurate pose estimation is a fundamental requirement not only for navigation and monitoring purposes, but also for control: within the considered system architecture, the robot motion is governed by a model-based control framework built around a Nonlinear Model Predictive Controller (NMPC). The NMPC relies on an estimate of the current robot state, including its pose, to predict the future evolution of the system and to compute dynamically consistent control actions.

The NMPC exploits a dynamic model of the quadruped that incorporates information from joint encoders, contact sensors, and the robot kinematics and dynamics and based on this model, the controller generates predictions of the robot motion and provides reference trajectories and inputs to a lower level whole-body controller (WBC). The information embedded in the dynamic model and from the contact sensors, represents a valuable source of information that can be exploited together with inertial measurements to support and constrain the state estimation process.

For this reason, the long-term perspective motivating this work is the development of a unified estimation framework in which inertial data, model-based information and data from other sensors are fused within an Extended Kalman Filter, without relying on a magnetometer. In the future, this framework can be naturally extended to include additional sources of information, such as SLAM-based corrections, to improve long-term consistency.

Within this context, the present work is structured as a progressive exploration of the following problems:

- First, the analysis of the capabilities and limitations of an IMU-only state estimation approach based on an Extended Kalman Filter, with the goal of gaining insight into the behaviour of inertial navigation in isolation.
- Second, the integration of information derived from the NMPC into the estimation framework, exploring different architectures and moving towards the final objective of achieving a stable and reliable pose estimation of the quadruped robot.

3.2. Hardware description

The inertial measurement unit (IMU) used in this work is the Gable ONE-series SE3 (Figure 3.1), a multi-sensor module, containing an XSENS MTi-1 series IMU with a 3-axis gyroscope, 3-axis accelerometer and 3-axis magnetometer. The MTi-1 series IMU is integrated in a fully functional EtherCAT slave which coordinates the communication between the EtherCAT network and the MTi-1.

Figure 3.1: Gable ONE-series SE3 in its HSD-version (Retrieved from [11]).

Together, these sensors allow not only to get the measurements of accelerations, angular velocities and magnetic field, but also an orientation estimate (quaternion/Euler angles) obtained through the internal sensor-fusion firmware. The device outputs acceleration and angular velocity in the sensor reference frame, while orientation information is supplied as a quaternion q_{ws} describing the rotation from the world frame (fixed) to the sensor frame.

In addition to estimating roll and pitch from the accelerometer and gyroscope fusion, the SE3 algorithm incorporates the magnetometer to obtain a yaw angle referenced to the

Earth's magnetic North (AHRS, Attitude and Heading Reference System). This allows the device to provide an almost drift-free orientation estimate in all three rotational degrees of freedom, unlike a pure IMU, where yaw remains unreferenced and is therefore affected by long-term drift.

According to the manufacturer datasheets, the gyroscope features an in-run bias stability of approximately 6 °/h and a noise density of 0.003 °/s/$\sqrt{\text{Hz}}$, while the accelerometer shows an in-run bias stability of about 40 μg and a noise density of 70 $\mu g/\sqrt{\text{Hz}}$. Regarding the buit-in orientation estimation, the SE3 AHRS provides a static roll and pitch accuracy of approximately 0.5° RMS and a dynamic yaw accuracy of about 2° RMS under typical operating conditions.

To address situations where the environment is subject to magnetic disturbances or distortions, the sensor includes the *Active Heading Stabilization* (AHS) feature. AHS modifies the behaviour of the heading estimator by suppressing the influence of the magnetometer on yaw determination whenever magnetic inconsistencies are detected; in this way, the yaw angle is no longer referenced to magnetic North but instead stabilizes around the startup orientation, significantly improving the estimate of orientation and reducing drift.

All the presented specifications and much more detailed information regarding the sensor, calibration procedures and filter behaviour can be found in the official datasheets [11, 34], provided by Gable Technologies and Xsens.

At the beginning of the project, a set of software tools was already available within the CRF framework. In particular, a dedicated IMU driver had been implemented to read the raw sensor data through EtherCAT, parse the incoming packets, and provide access to accelerations, angular velocities, and quaternions in real time. An EtherCAT communication layer was also available, handling low-level tasks such as bus synchronisation, device addressing, and cyclic data transfer, offering a stable acquisition infrastructure.

3.3. IMU sensor Characterization

To properly characterize the raw signals produced by the IMU (accelerations and orientations) it is essential to analyse their behaviour under controlled and repeatable conditions. Although datasheets provide nominal specifications for noise, bias stability, and accuracy, real sensors often exhibit additional effects due to thermal drift, calibration errors, non-idealities, or sensor-fusion artifacts. For this reason, a set of preliminary experiments were conducted, aimed at understanding the static behaviour of the IMU and identifying the main sources of error affecting the measurements.

Specifically, 60 minutes of data at 100 Hz were acquired , keeping the IMU completely stationary on a table. Two datasets were collected: one immediately after powering up the IMU ("cold-start"), and one after allowing the sensor to warm up for one hour ("hot-start"). These experiments allow the identification of long-term drifts, temperature-related transients, background noise, orientation errors, and residual bias components, providing a baseline for interpreting the IMU's behaviour during dynamic operation.

3.3.1. Cold-start and Hot-start comparison

A visual comparison of the raw acceleration signals in the sensor frame between the two datasets (cold-start and hot-start) revealed that the cold-start dataset exhibits a long transient, during which the measured accelerations drift over time (Figures 3.2 and 3.3).

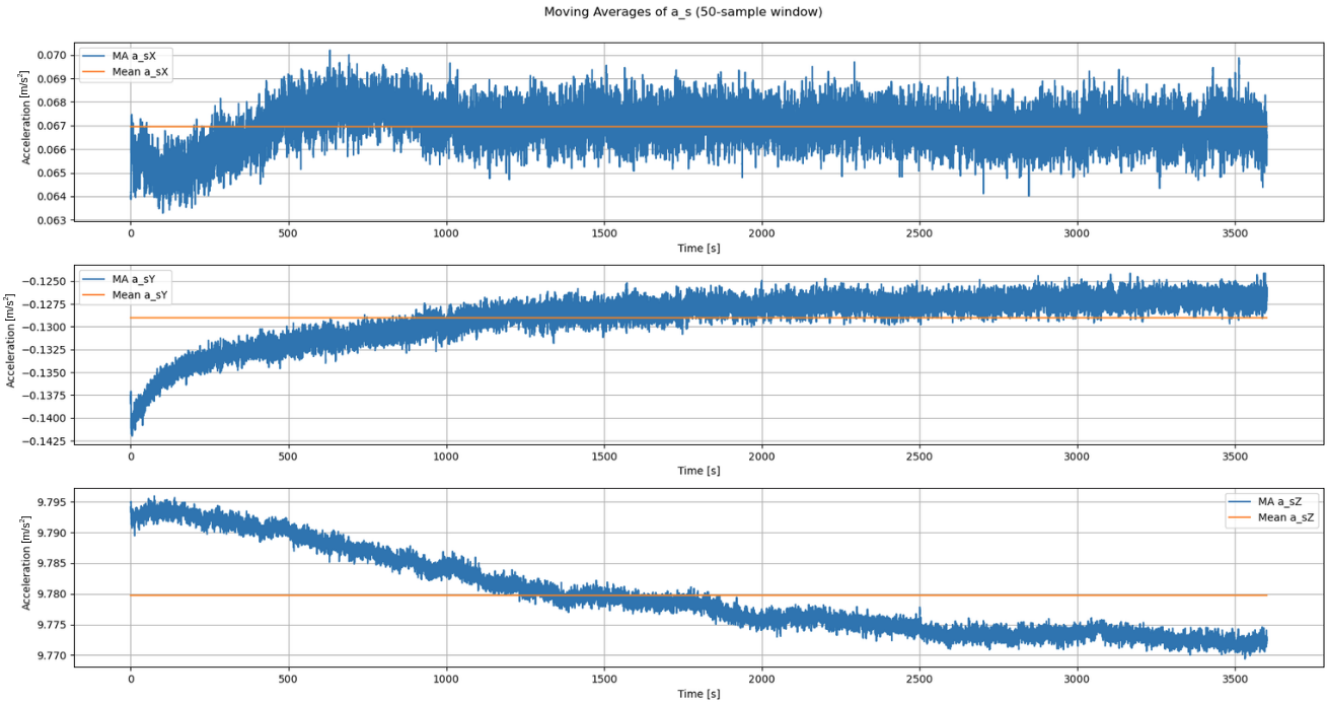

Figure 3.2: Cold-start experiment: acceleration in sensor frame filtered with a moving average for better visualization.

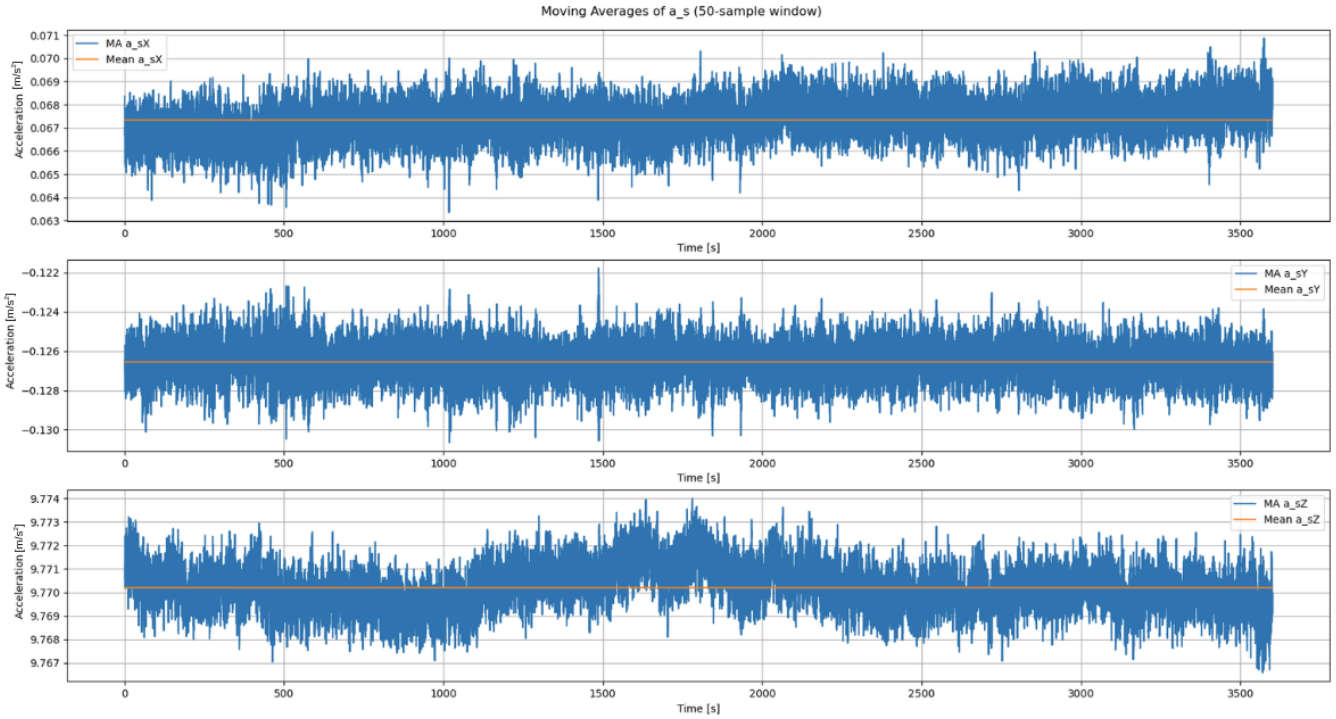

Figure 3.3: Hot-start experiment: acceleration in sensor frame filtered with a moving average for better visualization.

While the x and y axes appear to stabilise after approximately 30–40 minutes, the z-axis acceleration does not reach apparent stability even after one hour. This prolonged period of drift, varying in magnitude and direction, is likely attributable to thermal effects caused by the warm-up of the electronic components inside the IMU (power dissipation). This phenomenon is well documented in literature as we can see in [19–21]. For a more quantitative assessment of the difference between the two startup conditions, the maximum range of variation for each axis was computed for both cases and reported both in absolute terms and as a percentage with respect to the mean value (Table 3.1), confirming the results observed in the plots.

	MAX Δ [m/s²]			**MAX Δ [%]**		
	ΔX	ΔY	ΔZ	ΔX	ΔY	ΔZ
Cold start	0.0069	0.0179	0.0266	10.31	13.87	0.27
Hot start	0.0075	0.0089	0.0074	11.14	7.03	0.076

Table 3.1: Maximum variation of raw accelerations over one hour for cold and hot start conditions. Absolute variations and corresponding percentage variations with respect to the mean value.

A shorter initial transient in the cold-start dataset can also be observed in the orientation

estimate given by the sensor (Figures 3.4 and 3.5), probably due to the same reasons.

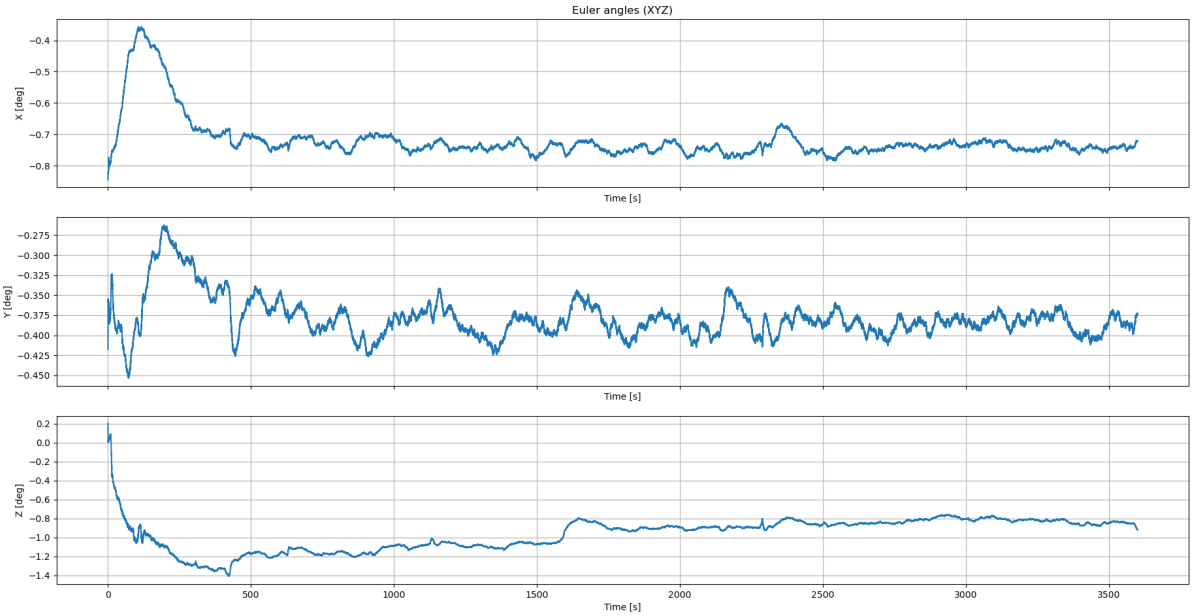

Figure 3.4: Cold-start experiment: Euler angles

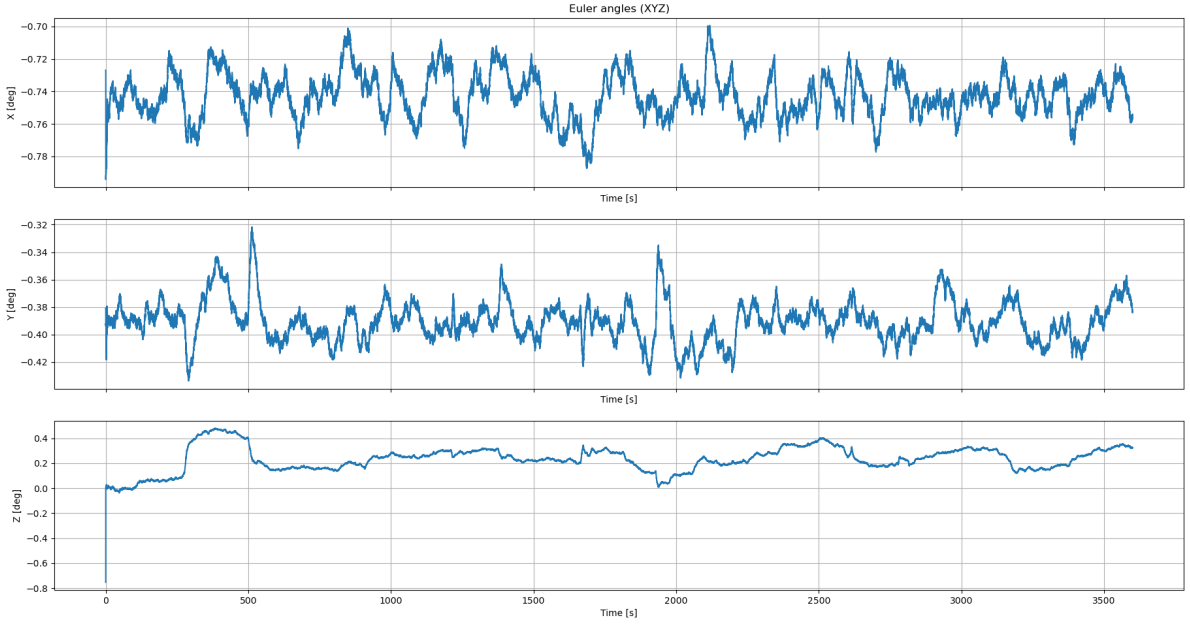

Figure 3.5: Hot-start experiment: Euler angles

3.3.2. Static background noise

Focusing the interest on the more stable hot-start dataset, the first parameter that can be computed is the magnitude of the background noise affecting the accelerometer read-

ings. Since the IMU remained perfectly stationary during the entire acquisition, any fast fluctuation in the measured accelerations can be attributed to measurement noise. For this reason, the standard deviation of each acceleration component provides a direct and meaningful estimate of the sensor noise level.

Let $a_{s,x}(t), a_{s,y}(t), a_{s,z}(t)$ denote the raw accelerations measured in the sensor frame. The noise for each axis is computed by removing the mean value and evaluating the standard deviation:

$$\sigma_x = \mathrm{STD}(a_{s,x}(t) - \overline{a}_{s,x}) \,, \tag{3.1}$$

$$\sigma_y = \mathrm{STD}(a_{s,y}(t) - \overline{a}_{s,y}) \,, \tag{3.2}$$

$$\sigma_z = \mathrm{STD}(a_{s,z}(t) - \overline{a}_{s,z}) \,, \tag{3.3}$$

where $\overline{a}_{s,i}$ denotes the mean of axis i. These values represent the background noise amplitude for each accelerometer axis under stationary conditions and can be visualized in the table below.

	Background noise amplitude
σ_x	0.000751
σ_y	0.000804
σ_z	0.000887

Table 3.2: Background noise amplitude on accelerations [m/s^2].

Estimating the noise offers an empirical indication of the signal level that should be expected during normal operation, which is crucial when tuning the process and measurement noise matrices Q and R of the Kalman filter. A high noise magnitude may also reveal additional sources of disturbance, such as vibrations.

3.3.3. Accelerometer Bias

In addition to the background noise, another important characteristic of the raw accelerometer measurements is the presence of a *bias* component. In the context of inertial sensors, the *bias* denotes an error affecting the measured acceleration defined in the sensor reference frame and generally different for each axis; it can also wander and change in time (normally slowly). Bias estimation and compensation is a critical aspect of inertial navigation since even small bias components, if not properly compensated, accumulate rapidly through integration and lead to significant velocity and position drift.
Ideally, the accelerometer should measure only the gravitational acceleration when sta-

tionary, with no additional constant term, however, the actual measurement is typically described by:

$$a_s(t) = R_{ws}(t)\, g_w + b_a + n_a(t),$$

where $R_{ws}(t)$ is the rotation matrix from world reference frame to sensor reference frame (associated with the orientation quaternion), g_w is the gravity vector expressed in the world frame, b_a is the accelerometer bias in the sensor frame, and $n_a(t)$ represents measurement noise.

The accelerometer bias ideally represents a hardware dependent offset added by each sensing axis, however in practice, isolating this "pure" bias term is extremely challenging, and the apparent offset observed in the gravity-free acceleration signal incorporates the combined effect of multiple error sources.

The gravity-free acceleration is obtained by removing the gravity vector projected into the sensor frame:

$$a_{s,\mathrm{nog}}(t) = a_s(t) - R_{sw}(t)\, g_w. \tag{3.4}$$

Any nonzero mean or slow variation in $a_{s,\mathrm{nog}}(t)$ reveals the presence of one or more bias related effects.

In general, the apparent bias visible in $a_{s,\mathrm{nog}}(t)$ may include contributions from:

- true hardware bias of each accelerometer axis,
- small errors in roll and pitch estimates, which cause gravity to leak onto the transverse axes,
- imperfect gravity compensation due to the IMU measuring a slightly different gravity norm than the nominal $9.81\,\mathrm{m/s^2}$,
- temperature-dependent variations in accelerometer and gyroscope behaviour,
- disturbances affecting the orientation estimator (e.g., magnetic interference, internal sensor fusion artifacts).

In particular, orientation errors play an important role. Since the gravity vector has a magnitude of $9.81\,\mathrm{m/s^2}$, even very small roll or pitch deviations result in a significant apparent acceleration component. For small angles, the leakage of gravity caused by an orientation error $\Delta\theta$ can be approximated as:

$$\Delta a \approx g \sin(\Delta\theta) \simeq g\, \Delta\theta,$$

with $\Delta\theta$ expressed in radians. Thus, an orientation error of only 0.1° ($1.745 \times 10^{-3}\,\mathrm{rad}$)

produces an apparent acceleration offset of:

$$\Delta a \approx 9.81 \times 1.745 \times 10^{-3} \approx 0.017\,\mathrm{m/s^2},$$

which is orders of magnitude larger than the intrinsic accelerometer noise. Considering for example an interval of 10 s, this Δa will produce a drift in position of 0.85 m.

This illustrates why bias is extremely sensitive to orientation accuracy, and generally how even a small bias could create important drifts in the estimated position .

Another way of spotting an orientation error in this static analysis is by comparing the 2 plots of the Euler angles and of the accelerations after having removed gravity ($a_{s,\mathrm{nog}} = a_s + R_{sw}\, g_w$), in particular for the cold start case. By comparing the Euler angles in Figure 3.4 with the gravity-compensated accelerations in Figure 3.6, a clear correlation can be observed. In particular, the temporal evolution of the roll angle exhibits the same shape, but with opposite sign, as the acceleration component along the Y-axis. A similar correspondence can be observed between the pitch angle and the acceleration along the X-axis. Since the orientation estimate is directly used to remove the gravity contribution from the measured accelerations, small variations or drifts in roll and pitch (which should ideally remain constant under static conditions) result in a spurious projection of the gravity vector onto the horizontal axes. As a consequence, apparent acceleration offsets are introduced, even though the sensor is physically stationary.

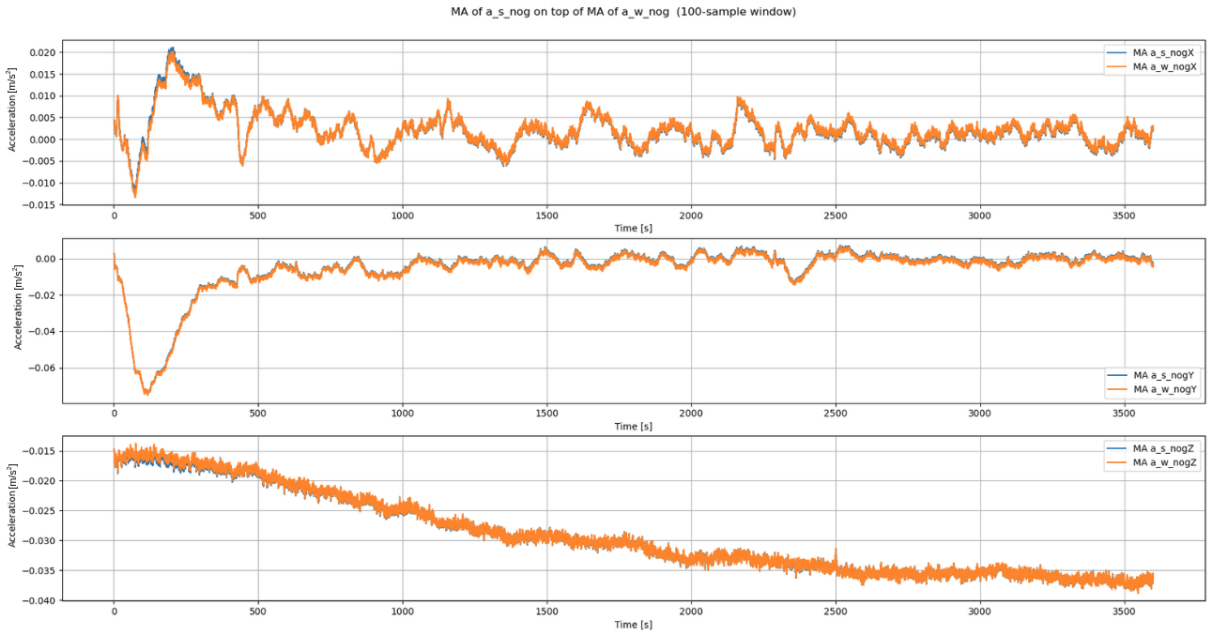

Figure 3.6: $a_{s,\mathrm{nog}}$ and $a_{w,\mathrm{nog}}$ plotted with a moving average of 50 samples for better visualization

Correctly distinguishing different sources of errors in bias can be really useful to design a perfect compensation or modelling strategy; however, it is not always fully achievable, especially when relying exclusively on the IMU's own quaternion estimates.

3.4. Position Estimation from IMU Data Using a Kalman Filter

Kalman filtering provides a well established probabilistic framework for estimating the state of a dynamic system by recursively combining a process model (system model) with noisy measurements. In the context of inertial navigation, this approach is particularly well suited, as IMU-based systems naturally rely on a predictive model derived from kinematic equations and on sensor measurements, affected by noise, bias, and modeling uncertainties. The Kalman Filter allows these effects and errors to be explicitly modelled and mitigated, maintaining both a state estimate and its associated uncertainty. One of the key advantages of the Kalman Filter is its recursive formulation: it continuously propagates past information forward in time, allowing previously accumulated errors and uncertainties to be taken into account when correcting future state estimates. For these reasons, Kalman-based estimators have become a standard tool in inertial navigation and robotics applications and are widely adopted in the literature.

Of course some fundamental limitations remain, mainly related to the achievable modeling accuracy, to the amount of available information (external measurements) and to the observability of the system. Nevertheless, it provides a structured and systematic way to incorporate and fuse the available information.

The last strength of this framework is its modularity and extensibility. The state vector, system model, and measurement equations can be easily extended to include additional sensors, constraints, or more refined error models. This makes the Kalman Filter particularly suitable as a baseline estimation framework, which can evolve from an IMU-only configuration to more complex multi-sensor fusion approaches as additional information becomes available.

This concept is reflected in the structure of the following sections, where first a general overview of the Extended Kalman Filter is provided, then moving to a detailed description of an IMU-only based approach for state estimation, its purpose and limitations.

It is worth to mention that in the inertial navigation literature, another common Kalman filtering formulation is the one based on an error-state Extended Kalman Filter (ES-EKF), [29, p. 29–36] , in which the filter estimates small errors around a nominal trajectory rather

than the full state directly. Consequently the error state must be injected in the nominal state at every iteration and then reset to zero to avoid accumulation of the same error and divergence of the estimate. This approach is especially used for attitude estimation, as it estimates small orientation errors instead of the orientation itself. When orientation is included as a state in a standard EKF, the update step applies additive corrections that are not naturally compatible with the unit-norm constraint of quaternions. By instead estimating the orientation error and applying corrections through incremental rotations, the error-state formulation preserves the validity of the orientation representation and simplifies both the filter structure and its numerical behaviour. Since a reliable orientation estimate is already available from the internal IMU fusion algorithm, this strategy is not pursued further in this chapter.

3.4.1. General formulation of EKF

Many state estimation problems including the inertial navigation one, involve nonlinear system dynamics and nonlinear measurement models. While the standard Kalman Filter provides an optimal recursive estimator for linear systems with Gaussian noise, it cannot be directly applied when the system or observation equations are nonlinear. In such cases, the Extended Kalman Filter (EKF) represents a widely adopted solution, as it allows nonlinear models to be handled.

The EKF approximates the nonlinear system dynamics and measurement functions by linearizing them around the current state estimate. This approach enables the use of the Kalman filtering framework while maintaining computational efficiency and real-time applicability, which is particularly important for inertial navigation systems.

System Model

Consider a discrete-time nonlinear system described by:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}, \tag{3.5}$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{n}_k, \tag{3.6}$$

where:

$\mathbf{x}_k$: state vector

$\mathbf{u}_k$: input vector

$\mathbf{z}_k$: measurement vector

$f(\cdot)$: function representing the nonlinear system dynamics

$h(\cdot)$: function representing the nonlinear measurement model

$\mathbf{w}_k$: process noise, $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, Q_k)$

$\mathbf{n}_k$: measurement noise, $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, R_k)$

With the process noise $\mathbf{w}_k$ and measurement noise $\mathbf{n}_k$ assumed to be zero-mean Gaussian with covariance matrices Q_k and R_k.

Prediction Step

At each time step, the EKF propagate the state estimate (finds the state prediction for the current time step) using the nonlinear system model:

$$\hat{\mathbf{x}}_k = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}). \tag{3.7}$$

The corresponding covariance prediction P_k is obtained by linearizing the system model around the current estimate:

$$P_k = F_{k-1} P_{k-1} F_{k-1}^\top + Q_{k-1}, \tag{3.8}$$

where F_{k-1} is the Jacobian (partial derivatives) of $f(\cdot)$ with respect to the state, evaluated at $\hat{\mathbf{x}}_{k-1}$. The covariance serves as a quantitative representation of the uncertainty associated with the state estimate. In practical terms, it encodes how reliable the predicted state is and how errors in one state component may affect the others. As the filter propagates forward in time, the covariance grows to reflect the accumulation of uncertainty due to model inaccuracies and sensor noise. This information directly influences the correction step, governing the strength of the update and the coupling between state variables.

Measurement Update

When a measurement is available, the EKF performs the correction step. The innovation is computed as:

$$\mathbf{y}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_k), \tag{3.9}$$

with innovation covariance:

$$S_k = H_k P_k H_k^\top + R_k, \tag{3.10}$$

where: $\mathbf{H}_k$ is the Jacobian of the measurement function $h(\cdot)$ with respect to the state.

The Kalman gain is then given by:

$$K_k = P_k H_k^\top S_k^{-1} = P_k H_k^\top \left(H_k P_k H_k^\top + R_k\right)^{-1}. \tag{3.11}$$

Finally, the state and covariance estimates are updated using the computed Kalman gain as:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k + K_k \mathbf{y}_k, \tag{3.12}$$

$$P_k = (I - K_k H_k) P_k. \tag{3.13}$$

Although the linearization introduces approximations, the EKF has proven effective for a wide range of inertial navigation and robotics applications, when the system operates close to the linearization point.

3.4.2. ZUPT-based correction EKF formulation

In this section, the Extended Kalman filter presented previously will be used trying to exploit just the information from the IMU sensor for estimating its position. The objective of this approach is to investigate the intrinsic capabilities and limitations of IMU-only state estimation, establishing a baseline estimation pipeline that can be expanded in future developments.

While in literature it is common to start from raw data of accelerometer, gyroscope and magnetometer sensors, estimating also the orientation from scratch, here an hybrid strategy is adopted. The orientation provided by the IMU is used directly as an input to the filter and assumed to be sufficiently accurate, and the EKF is employed to estimate position, velocity, and accelerometer bias. This design choice is motivated by the fact that estimating orientation within the EKF would require explicit modelling of gyroscope bias, careful handling of the nonlinear attitude dynamics, and the development of the right signal pre-processing and sensor fusion algorithms to use gyroscope and magnetometer

measurements. Other then that, the IMU already provides a reliable orientation estimate through its onboard sensor-fusion algorithm and directly using this information allows avoiding unnecessary model complexity.

The main drawback of this strategy is the reduced amount of information available for correcting the state estimate. By relying on the IMU-provided orientation and avoiding the direct use of gyroscope measurements within the EKF, the filter cannot rely on the gyroscope to correct orientation related errors. As a result, the estimator becomes more dependent on the system model. It makes it also difficult to distinguish between different error sources, leading to poorly observable error states, not really usable for the purpose of making a more detailed system model.

These considerations also motivated the choice of modelling only an accelerometer bias term within the EKF. In the IMU-only configuration considered here, the limited amount of independent information available for correction makes it difficult to reliably distinguish between multiple error sources, such as orientation errors, gravity mismodeling, or temperature dependent errors. For this reason, rather than introducing several weakly observable error states, the combined effect is absorbed into a single accelerometer bias term. In the literature, accelerometer bias is commonly modelled as a slowly varying quantity, often represented as a Gaussian random walk. This modelling choice is adopted in the present approach, with the only adjustment being the tuning of the corresponding process noise covariance in the Q matrix to allow relatively fast bias adaptation. In this way, the bias term is able to absorb both low and high frequency error components, effectively compensating for multiple sources of drift.

Correction strategy: ZUPT

As seen before, inertial-only state estimation suffers from unbounded error growth due to noise and bias accumulation. To limit this drift, the Kalman Filter requires corrective information in the form of measurements or constraints, however, when only IMU data are available and no external sensors are used, the amount of information that can be exploited for correction is extremely limited. One of the few sources of exploitable information in this scenario arises from stationary phases of the system. During these intervals, the true velocity of the platform is known *a priori* to be zero; this knowledge can be introduced into the estimation process as a pseudo-measurement, providing a constraint that can be used to correct the predicted state. This approach is commonly referred to as a Zero-Velocity Update (ZUPT) and represents a well-established technique in inertial navigation literature [10, 18, 32].
In a ZUPT-aided inertial navigation system, when the system is detected to be stationary,

the estimated velocity, which may contain accumulated errors, is forced to zero through a Kalman Filter update (more or less aggressively depending on Q). Outside the stationary intervals, the filter operates in prediction mode using only inertial measurements and the system model (double integration). In this context, velocity is the only state that is directly observable and therefore directly corrected by the zero-velocity measurement [18]. Position and accelerometer bias, on the other hand, are not directly observable during stationary phases but they can be corrected indirectly exploiting their coupling with velocity in the system dynamics (system model). The information that velocity is influenced by accelerometer bias and position is influenced by velocity is stored in the state covariance matrix (cross-covariance terms), and enables to propagate the Kalman Filter update to these 2 states.

A key limitation of this method is the inability to consistently track position over more than a few seconds without external corrections (motion pauses), which makes it suitable only for motions that include frequent stationary phases. Although bias estimation mitigates position drift during motion, any variation of the bias or unmodeled errors during a movement phase cannot be corrected until a subsequent stationary phase occurs. For these reasons ZUPT is one of the main correction methods used in pedestrian tracking systems, where a pause occurs at every step.

The effectiveness of the ZUPT-based correction strategy also depends on the reliability of the detection of stationary phases. (An incorrect classification of motion or rest can lead to either missed correction opportunities or erroneous updates, both of which negatively affect estimation performance). For this reason, a dedicated stationary detection algorithm is required; detailed in the following section.

Stationary detector logic

The stationary detector aim is to identify time intervals in which the robot is not moving. This problem has been widely studied in the context of zero-velocity-aided inertial navigation, where it is commonly addressed using threshold-based detectors or likelihood-ratio-test (LRT) formulations built on accelerometer and gyroscope signals [27].

On a real robot, possible sources of stationarity information could in principle be inferred from leg odometry, joint kinematics, foot contact sensors or other internal signals. However, to keep the estimator self-contained and to evaluate the capabilities of IMU-only state estimation in a simulation environment, stationarity is detected here using only IMU measurements.

The adopted stationary detector is a threshold-based method that evaluates the magni-

tude of gravity-compensated acceleration and angular velocity over a short sliding window of N samples. The underlying assumption is that, during stationary phases, both acceleration and angular rate remain close to zero.

Let $a_{nog}(k)$ denote the gravity-free acceleration, and $\omega(k)$ denote the measured angular velocity. Given the window $\mathcal{W}_k = \{k - N + 1, \ldots, k\}$ of N consecutive samples, the following root-mean-square (RMS) quantities are computed:

$$A_k = \sqrt{\frac{1}{N} \sum_{i \in \mathcal{W}_k} \|a_{w,\text{nog}}(i)\|^2}, \tag{3.14}$$

$$\Omega_k = \sqrt{\frac{1}{N} \sum_{i \in \mathcal{W}_k} \|\omega(i)\|^2}. \tag{3.15}$$

The system is classified as stationary at time k when both conditions

$$A_k < \gamma_a \quad \text{and} \quad \Omega_k < \gamma_\omega \tag{3.16}$$

are satisfied. The thresholds γ_a and γ_ω are tuned based on the characteristics of the specific operating conditions, including sensor noise levels, expected vibrations, and motion dynamics.

Using a windowed statistic instead of just looking at the instantaneous values reduces sensitivity to outliers and high-frequency artifacts, and helps preventing false detections. At the same time, increasing N introduces detection delay, which may be a problem; for this reason N should be properly tuned to achieve a trade-off between robustness (against false stationary detections) and responsiveness. Avoiding false detections is particularly important, as they introduce an incorrect zero-velocity constraint into the filter. To avoid this issue conservative threshold selection and windowing strategies are generally preferred, even at the cost of reducing the number of ZUPT updates.

In theory, the explained method may fail in specific edge cases, such as near-constant velocity motion along a straight trajectory with very low angular rates and limited variation in the measured specific force. However, in practical scenarios, this limitation is mitigated by the presence of small vibrations, impacts, or residual angular motion, which typically make stationary and moving phases distinguishable when combining accelerometer and gyroscope information.

Discrete-Time EKF Formulation for IMU-Based State Estimation

The general EKF mathematical formulation described previously will be adapted here for the specific uses case, following all the constraints and needs of the adopted IMU-only based estimation with ZUPT-based corrections. This section will not only aim at showing the detailed theoretical formulation including state definition, system model, and measurement model, but it will also motivate some design choices and try to give a more practical and intuitive view of how the Kalman filter works.

State vector definition:
The chosen state vector for the EKF is:

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{p}(k) \\ \mathbf{v}(k) \\ \mathbf{b}_a(k) \end{bmatrix}, \tag{3.17}$$

where $\mathbf{p}(k) \in \mathbb{R}^3$ represents the position of the robot, $\mathbf{v}(k) \in \mathbb{R}^3$ its linear velocity, and $\mathbf{b}_a(k) \in \mathbb{R}^3$ the accelerometer bias, all expressed in the reference frame used by the estimator at discrete time step k.

This modelling choice is motivated by the fact that, in inertial-only navigation, the dominant source of drift in velocity and position estimates originates from biases errors in the measured accelerations. In practice, the apparent acceleration bias may arise from multiple sources, hardware-based bias of the sensor, orientation errors, gravity mismodelling, and temperature-dependent effects. However, the amount of available information is generally insufficient to reliably distinguish and separately estimate these different error contributions. For this reason, rather than explicitly modeling each individual error source, their combined effect is captured through one single term.

Process model:
The EKF is driven by IMU measurements and by the orientation provided by the IMU sensor-fusion algorithm and it combines these information using some equations to predict the states of the system. Let $\mathbf{a}_s(k)$ denote the raw acceleration measured in the sensor frame, and let $R_{ws}(k)$ be the rotation matrix from sensor to world frame (obtained from the IMU orientation). The gravity-compensated acceleration in the world frame is computed as

$$\mathbf{a}_w(k) = R_{ws}(k)\,(\mathbf{a}_s(k) - \mathbf{b}_a(k)) + \mathbf{g}, \tag{3.18}$$

where $\mathbf{g_w}$ is the gravity vector expressed in the world frame.

Using a discrete-time kinematic model with sampling time Δt, the state propagation equations are given by

$$\mathbf{v}(k) = \mathbf{v}(k-1) + \mathbf{a}_w(k)\,\Delta t, \tag{3.19}$$

$$\mathbf{p}(k) = \mathbf{p}(k-1) + \mathbf{v}(k)\,\Delta t + \tfrac{1}{2}\mathbf{a}_w(k)\,\Delta t^2, \tag{3.20}$$

$$\mathbf{b}_a(k) = \mathbf{b}_a(k-1) + \mathbf{w}_b(k), \tag{3.21}$$

where $\mathbf{w}_{b,k}$ is a zero-mean Gaussian noise.

The first two equations (3.19) and (3.20) describe the discrete-time integration of the measured acceleration to obtain velocity and position, and explicitly show how errors in the acceleration measurements propagate to the remaining states. The accelerometer bias is modelled as a Gaussian random walk, as it is often done in literature, considering it as a slow changing pure hardware-based bias. As was already stated the bias term incorporates in reality different errors and contributions, many of which are weakly observable or completely unobservable when relying on inertial measurements alone. Even if modeling bias simply as a random walk is not theoretically correct, it has been shown [18] to be effective in practice when frequent zero-velocity updates are available, making it unnecessary to create a more detailed and complex model. This modelling choice also simplifies the filter structure and tuning process by limiting the number of states and parameters. Assigning a relatively large process noise to the bias state encourages the filter to absorb unmodelled errors into the bias term, also allowing it to adapt more quickly to changing conditions.

Prediction step

At each time step, the EKF performs the **prediction step** by using the system equations:

- Propagating the state estimate and
- updating its associated covariance.

Since the system model is nonlinear, the Jacobian matrix F_k is computed by linearizing the process model around the current state estimate (using the previously stated equations). The resulting Jacobian matrix is:

$$F_k \triangleq \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k|k}} = \begin{bmatrix} I_3 & \Delta t\, I_3 & -\frac{1}{2}\Delta t^2\, R_{ws}(k) \\ 0_3 & I_3 & -\Delta t\, R_{ws}(k) \\ 0_3 & 0_3 & I_3 \end{bmatrix}. \tag{3.22}$$

The Jacobian matrix provides insight into the interactions between the state variables; it make is easier to understand how corrections applied to the velocity during ZUPT updates indirectly affect the remaining states through covariance propagation.

The predicted covariance is then computed as before using the Jacobian matrix:

$$P_k = F_k P_{k-1} F_k^\top + Q_k, \tag{3.23}$$

P_k represents the uncertainty associated with the current state estimate, and Q_k is the process noise covariance matrix, that encodes the confidence in the system model, accounting for accelerometer noise and bias random walk. Q_k plays a key role in balancing model prediction against measurement correction, and has to be tuned to achieve the best result.

Measurement model and ZUPT update.
When the stationary detector indicates that the system is at rest, a zero-velocity update is applied. In this case, the measurement is just the velocity:

$$\mathbf{z}(k) = \mathbf{0}, \tag{3.24}$$

corresponding to the known condition that the true velocity is zero during stationary phases. As stated before it's not a real measurement but more of a pseudo-measurement deduced from the physics of the system in the stationary state.

The measurement model is therefore

$$\mathbf{z}(k) = h(\mathbf{x}(k)) + \mathbf{n}(k) = H\mathbf{x}(k) + \mathbf{n}(k), \tag{3.25}$$

$\mathbf{n}(k)$ representing measurement noise with covariance R and with

$$H_k \triangleq \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_k} = \begin{bmatrix}\mathbf{0}_{3\times3} & I_{3\times3} & \mathbf{0}_{3\times3}\end{bmatrix}. \tag{3.26}$$

where H is the Jacobian matrix of the measurement function $h(\cdot)$, reflecting the fact that the direct correction performed by the KF affects just the velocity state.

Correction step and Kalman gain.
The innovation now can be computed as the difference between the measured and the

predicted velocity:

$$\mathbf{y}(k) = \mathbf{z}(k) - h(\hat{\mathbf{x}}(k)) = \mathbf{0} - \hat{\mathbf{v}}(k) = -\hat{\mathbf{v}}(k); \tag{3.27}$$

along with the innovation covariance, computed as

$$S_k = H_k P_k H_k^\top + R, \tag{3.28}$$

where R is the measurement noise covariance associated with the zero-velocity constraint. Intuitively, R represents the confidence assigned to the assumption that the true velocity is zero during stationary phases, accounting for imperfections in the stationary detection. Then the Kalman gain:

$$K_k = P_k H^\top \left(H P_k H^\top + R\right)^{-1} \tag{3.29}$$

Intuitively, the Kalman gain determines the correction to be applied to the state estimate, based on the uncertainties contained in the covariance matrix; so basically based on how much the filter trusts the zero-velocity constraint.

Finally the state and covariance are updated as

$$\hat{\mathbf{x}}(k) = \mathbf{x}(k) + K_k \mathbf{y}(k), \tag{3.30}$$

$$P_k = (I - K_k H) P_k, \tag{3.31}$$

While the state update applies the correction to the current estimate, the covariance update quantifies the increased confidence in the estimate and determines how this information is shared among the different state components.

The strength and responsiveness of the correction applied during stationary phases are directly influenced by the measurement and process noise covariance matrices R and Q, which therefore require careful tuning.

3.5. State estimation for a quadruped robot exploring different fusion strategies

In this section the focus will shift to the pose estimation of a quadruped robot using an extended Kalman filter to fuse together multiple sources of information. The limitations observed in the IMU-only approach, in particular the strong dependence on frequent stationary phases for drift correction and the dependence on the magnetometer to obtain

the orientation estimate, highlighted the need for a more robust and long term estimation strategy when considering the actual application scenario of a quadruped robot.

The objective is now to investigate alternative strategies to estimate the position and also the orientation of a quadruped robot by exploiting not only inertial measurements but also the information available from the robot control architecture. The robot controller requires, at each control cycle, an estimate of the current robot state, including the pose of base frame, in order to predict the future evolution of the system and compute dynamically consistent control actions. As a consequence, state estimation is not only required for navigation and monitoring, but is a fundamental prerequisite for stable locomotion control.

Quadruped controller architecture

The quadruped is controlled using a model-based architecture composed of a Nonlinear Model Predictive Controller (NMPC) and a Whole-Body Controller (WBC). The NMPC operates at a higher level and exploits a dynamic model of the robot to solve an optimal control problem over a finite prediction horizon. Given the current estimated state, reference trajectories, and contact constraints, the NMPC generates dynamically consistent predictions of the robot motion, including the evolution of the base frame position, velocity and base orientation.
Due to computational constraints, the NMPC is solved asynchronously and at a lower rate than the low-level control loop. Its discrete solutions are interpolated in time and passed as references to the WBC. The WBC ensures that the planned trajectories are physically feasible and executed consistently by the robot, it is formulated as a quadratic program, and runs at a higher frequency enforcing full-body dynamics, contact constraints, and actuator limits, while tracking the motion references provided by the NMPC.

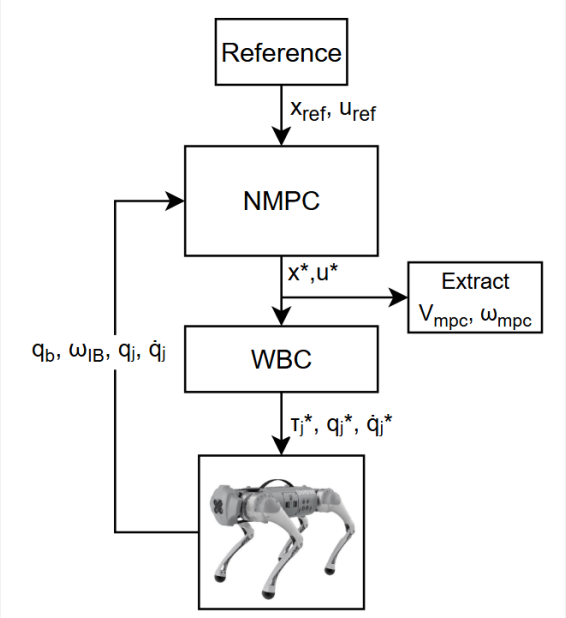

Figure 3.7: Diagram of the robot control architecture.

Figure 3.7 helps better understanding the general structure of the controller. The reference signals x_{ref} and u_{ref} define the desired motion objectives for the NMPC, including base and center of mass trajectories and nominal input profiles.
The optimal predicted state and input trajectories computed by the NMPC are x^* and u^* which are then provided to the whole-body controller. τ_j^* contains the desired joint torques while q_j^* and $\dot{q}_j^*$ the desired joint positions and velocities.
The NMPC needs the measured base pose (q_b) and base angular velocity (ω_{IB}) with respect to the inertial frame, and also the measured joint positions (q_j) and velocities ($\dot{q}_j$), acquired from joint encoders.
From the NMPC solution, the base linear and angular velocity references ($\mathbf{v}_{\text{mpc}}, \boldsymbol{\omega}_{\text{mpc}}$) can be extracted and possibly used for state estimation.

From a state estimation perspective, several sources of information are available on the robot and could in principle be exploited to improve pose estimation (for example joint positions and velocities from encoders and contact state information from foot sensors).

A common approach in the literature for legged robots consists of explicitly fusing IMU measurements with leg kinematics and contact constraints (e.g. [3]), building a state estimator that directly exploits encoder data and foot contact information to correct the estimate and limit the drift.

In the next subsections, however, a conceptually simpler intermediate step is explored. Instead of explicitly modeling leg kinematics and contact switching within the estimator, the focus is placed on directly exploiting the output of the NMPC itself. The NMPC already embeds the robot dynamic model and integrates encoder and contact information internally to generate predictions of the base motion. Its output (in particular the predicted base frame linear and angular velocities) can be interpreted as model-consistent motion estimates that may serve as additional information for state estimation, without requiring the implementation of a full leg odometry pipeline, limiting the modeling complexity.

The following sections analyse these two complementary architectures:

- an approach where inertial propagation is used as process model and NMPC-predicted velocities are introduced as corrective measurements, and
- a dual approach where the NMPC-based model is used as the main propagation mechanism while inertial measurements provide corrective information.

The purpose of this analysis is exploratory, aiming at understanding the interaction between inertial sensing and model-based controller predictions, identifying potential limitations, and motivating future developments.

3.5.1. IMU-model-based EKF with MPC corrections

The first explored fusion strategy preserves the structure of the IMU-only estimator, modifying only the correction mechanism. The idea is to rely on inertial measurement for the model propagation, while replacing the ZUPT correction with a measurement derived from the NMPC outputs.

The underlying intuition is aligned with the common practice in multi sensor fusion of using high-frequency measurements to propagate the state estimate, while exploiting lower-frequency information, when available, to correct the state estimate. IMU measurements, available at high rate and directly linked to the system dynamics, are therefore well suited for state propagation and widely used in literature for this purpose. The NMPC provides predictions of the base motion at a lower rate, which can be interpreted as correcting information to constrain the inertial estimate.

State

Unlike the IMU-only case, in this configuration it becomes necessary to estimate not only the position but also the orientation of the robot. Since the onboard IMU sensor-fusion

algorithm (which relies on the magnetometer) cannot be used here, the orientation must be explicitly included as a state variable of the filter. The EKF state therefore includes position, velocity, and orientation of the robot base. Accelerometer and gyroscope bias states are not considered, as the analysis is performed in simulation using an ideal IMU model without bias. If required in future developments, bias terms could be easily included by augmenting the state vector and introducing the corresponding coupling terms in the system model.

The state vector is therefore defined as:

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{p}(k) \\ \mathbf{v}(k) \\ \mathbf{q}(k) \end{bmatrix} \in \mathbb{R}^{10}, \tag{3.32}$$

where $\mathbf{p}(k) \in \mathbb{R}^3$ is the base position in the world frame, $\mathbf{v}(k) \in \mathbb{R}^3$ is the base linear velocity in the world frame, and $\mathbf{q}(k) = [q_w, q_x, q_y, q_z]^\top$ is the unit quaternion representing the rotation from body frame to world frame.

The orientation is represented here directly by the quaternion state. Although error-state approaches are often preferred for improved consistency on the rotation update pipeline, the direct quaternion formulation was adopted here to limit model complexity and state dimension. The consequences of this choice, particularly related to the KF correction update, are addressed in the Paragraph 3.5.1.

System model (IMU-based propagation)

The propagation step is based on the standard inertial navigation equations. Linear acceleration measurements, expressed in the body frame, are rotated into the world frame and integrated to update velocity and position, while angular velocity measurements are integrated to propagate the orientation.

It is important to note that, in a real scenario relying exclusively on gyroscope integration, doesn't make orientation fully observable in the absence of external heading references such as magnetometers, vision, or contact-based constraints. As a consequence, any bias or drift in the measured angular velocity around the vertical axis directly results in an accumulated error over time. In the present simulated scenario, an ideal IMU model is considered, and therefore gyroscope bias is neglected. Under this assumption, integrating angular velocity provides a consistent orientation estimate; however, in a real-world setup this approach would require the addition of a gyroscope bias state to prevent long-term drift.

The EKF prediction step uses the IMU measurements as control input:

$$\mathbf{u}(k) = \begin{bmatrix} \mathbf{a}_b(k) \\ \boldsymbol{\omega}_b(k) \end{bmatrix}, \tag{3.33}$$

where $\mathbf{a}_b(k)$ and $\boldsymbol{\omega}_b(k)$ are the linear acceleration and angular velocity measured by the IMU, expressed in the body frame.

The discrete-time system model is given by:

$$\mathbf{q}(k+1) = \mathbf{q}(k) \otimes \delta\mathbf{q}(\boldsymbol{\omega}_b(k)\,\Delta t)\,, \tag{3.34}$$

$$\mathbf{a}_w(k) = \mathbf{R}(\mathbf{q}(k))\ \mathbf{a}_b(k), \tag{3.35}$$

$$\mathbf{v}(k+1) = \mathbf{v}(k) + (\mathbf{a}_w(k) + \mathbf{g})\,\Delta t, \tag{3.36}$$

$$\mathbf{p}(k+1) = \mathbf{p}(k) + \mathbf{v}(k)\Delta t + \tfrac{1}{2}\,(\mathbf{a}_w(k) + \mathbf{g})\,\Delta t^2. \tag{3.37}$$

where $\mathbf{R}(\mathbf{q})$ is the rotation matrix associated with the quaternion $\mathbf{q}$, $\mathbf{g} = [0, 0, -9.81]^\top$ is the gravity vector expressed in the world frame, and $\otimes$ denotes quaternion multiplication. The incremental quaternion $\delta\mathbf{q}(\boldsymbol{\omega}\Delta t)$ is computed using the Euler–Rodrigues formula.

Linearization: computation of the Jacobian

Since the Extended Kalman Filter relies on a first-order linearization of the nonlinear system dynamics, the computation of the Jacobian matrix of the process model is a fundamental step of the prediction phase. While several terms of this matrix can be easily derived from the system model equations, the blocks involving the quaternion require a more careful derivation. For this reason, the following paragraphs detail the computation of these less straight forward Jacobian terms.

The Jacobian of the discrete-time system model with respect to the state is structured as

$$\mathbf{G}_k = \begin{bmatrix} \mathbf{I} & \Delta t\,\mathbf{I} & \mathbf{G}_{p,q} \\ \mathbf{0} & \mathbf{I} & \mathbf{G}_{v,q} \\ \mathbf{0} & \mathbf{0} & \mathbf{G}_{q,q} \end{bmatrix}, \tag{3.38}$$

where $\mathbf{G}_{q,q}$ corresponds to the Jacobian of the quaternion propagation term, while the blocks $\mathbf{G}_{p,q}$ and $\mathbf{G}_{v,q}$ arise from the dependence of the rotated acceleration $\mathbf{a_w} = \mathbf{R}(\mathbf{q})\,\mathbf{a}_b$ on the quaternion.

To simplify the derivations, the quaternion is decomposed into scalar and vector parts as

$$\mathbf{q} = \begin{bmatrix} s \\ \mathbf{u} \end{bmatrix}, \qquad s \in \mathbb{R},\ \mathbf{u} \in \mathbb{R}^3, \tag{3.39}$$

with the unit-norm constraint $\|\mathbf{q}\| = 1$.

The quaternion propagation depends on the body angular velocity $\boldsymbol{\omega}_b$ and it is performed as shown in 3.34.

To express this operation in linear form with respect to $\mathbf{q}_k$, we introduce the right quaternion multiplication matrix $\mathbf{R}(\delta\mathbf{q}_k) \in \mathbb{R}^{4\times4}$, defined such that

$$\mathbf{q}_k \otimes \delta\mathbf{q}_k = \mathbf{R}(\delta\mathbf{q}_k)\,\mathbf{q}_k. \tag{3.40}$$

With this notation, the Jacobian of the quaternion propagation with respect to the state quaternion is directly obtained as

$$\mathbf{G}_{q,q} = \mathbf{R}(\delta\mathbf{q}_k) = \begin{bmatrix} \delta s & -\delta u_x & -\delta u_y & -\delta u_z \\ \delta u_x & \delta s & \delta u_z & -\delta u_y \\ \delta u_y & -\delta u_z & \delta s & \delta u_x \\ \delta u_z & \delta u_y & -\delta u_x & \delta s \end{bmatrix}, \tag{3.41}$$

where the incremental quaternion is written as $\delta\mathbf{q}_k = [\delta s,\ \delta\mathbf{u}]^\top$.

To compute $\mathbf{G}_{p,q}$ and $\mathbf{G}_{v,q}$ it is convenient to introduce a quaternion identity to rotate a vector $\mathbf{a}_b$ (the rotation identity and the associated quaternion algebra are consistent with [16] p. 45):

$$\boxed{\mathbf{R}(\mathbf{q})\,\mathbf{a}_b = \left(s^2 - \mathbf{u}^\top\mathbf{u}\right)\mathbf{a}_b + 2\,\mathbf{u}\left(\mathbf{u}^\top\mathbf{a}_b\right) + 2s\,(\mathbf{u}\times\mathbf{a}_b)}. \tag{3.42}$$

This form makes the differentiation with respect to $\mathbf{s}$ and $\mathbf{u}$ explicit making it possible to split the derivative with respect to q in 2 parts:

$$\mathbf{J}_s = \frac{\partial\mathbf{a}_w}{\partial s} \qquad \text{and} \qquad \mathbf{J}_u = \frac{\partial\mathbf{a}_w}{\partial\mathbf{u}}$$

Differentiating (3.42) with respect to $\mathbf{s}$ we obtain:

$$\frac{\partial}{\partial s}\left((s^2 - \mathbf{u}^\top \mathbf{u})\mathbf{a}_b\right) = 2s\,\mathbf{a}_b, \tag{3.43}$$

$$\frac{\partial}{\partial s}\left(2\,\mathbf{u}(\mathbf{u}^\top \mathbf{a}_b)\right) = \mathbf{0}, \tag{3.44}$$

$$\frac{\partial}{\partial s}\left(2s(\mathbf{u} \times \mathbf{a}_b)\right) = 2(\mathbf{u} \times \mathbf{a}_b); \tag{3.45}$$

that composing the various parts becomes:

$$\boxed{\mathbf{J}_s = 2s\,\mathbf{a}_b + 2(\mathbf{u} \times \mathbf{a}_b)} \in \mathbb{R}^{3\times 1}. \tag{3.46}$$

Again, differentiating (3.42), now with respect to $\mathbf{u}$:

$$\frac{\partial}{\partial \mathbf{u}}\left(-(\mathbf{u}^\top \mathbf{u})\mathbf{a}_b\right) = -2\,\mathbf{a}_b\,\mathbf{u}^\top. \tag{3.47}$$

$$\frac{\partial}{\partial \mathbf{u}}\left(2\,\mathbf{u}(\mathbf{u}^\top \mathbf{a}_b)\right) = 2\left[\frac{\partial \mathbf{u}}{\partial \mathbf{u}}(\mathbf{u}^\top \mathbf{a}_b) + \mathbf{u}\frac{\partial(\mathbf{u}^\top \mathbf{a}_b)}{\partial \mathbf{u}}\right] = 2\left[(\mathbf{u}^\top \mathbf{a}_b)\mathbf{I} + \mathbf{u}\mathbf{a}_b^\top\right]. \tag{3.48}$$

And using the identity $\mathbf{u} \times \mathbf{a}_b = -[\mathbf{a}_b]_\times\,\mathbf{u}$, the last term is obtained:

$$\frac{\partial}{\partial \mathbf{u}}\left(2s(\mathbf{u} \times \mathbf{a}_b)\right) = -2s\,[\mathbf{a}_b]_\times. \tag{3.49}$$

Summing all contributions yields

$$\boxed{\mathbf{J}_u = 2(\mathbf{u}^\top \mathbf{a}_b)\mathbf{I} + 2\,\mathbf{u}\mathbf{a}_b^\top - 2\,\mathbf{a}_b\mathbf{u}^\top - 2s\,[\mathbf{a}_b]_\times} \in \mathbb{R}^{3\times 3}. \tag{3.50}$$

Lastly it is possible to assemble the final Jacobian matrix as:

$$\boxed{\mathbf{J}_q = \frac{\partial \mathbf{a}_w}{\partial \mathbf{q}} = \begin{bmatrix}\mathbf{J}_s & \mathbf{J}_u\end{bmatrix}} \in \mathbb{R}^{3\times 4} \tag{3.51}$$

and compute the 2 terms $\mathbf{G}_{p,q}$ and $\mathbf{G}_{v,q}$; remembering the initial system equations 3.36,

3.37 and following the chain rule:

$$\boxed{\mathbf{G}_{v,q} = \Delta t\ \mathbf{J}_q, \qquad \mathbf{G}_{p,q} = \tfrac{1}{2}\Delta t^2\ \mathbf{J}_q}. \tag{3.52}$$

Observation model (MPC-based velocity correction)

In this formulation, the correction step is performed continuously using the NMPC predicted velocity. Unlike the IMU-only approach, where corrections are applied only during detected stationary phases (ZUPT), here the update is carried out at each control cycle, or more generally at a constant rate determined by the availability of the MPC output. The core idea is to exploit the model-based velocity estimate, provided by the controller as an external measurement of the robot base velocity expressed in the world frame. This correction should help constraining the inertial propagation, reducing the velocity drift and indirectly also correcting position and orientation through the coupling terms in the covariance matrix. Let $\mathbf{v}_{\text{mpc}}(k)$ denote the linear velocity of the base predicted by the NMPC at time step k.

The measurement vector is therefore defined as

$$\mathbf{z}(k) = \mathbf{v}_{\text{mpc}}(k), \tag{3.53}$$

while the observation function extracts the velocity component from the state:

$$\mathbf{h}(\mathbf{x}(k)) = \mathbf{v}(k). \tag{3.54}$$

The corresponding observation Jacobian is constant and given by

$$\mathbf{H} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times4} \end{bmatrix}. \tag{3.55}$$

It is important to note that no direct orientation correction is applied in this architecture and the quaternion state is therefore updated only indirectly through the Kalman gain coupling with the velocity correction.

Since the quaternion belongs to the unit sphere $\mathbb{S}^3$, the EKF correction step formally

applies an additive update:

$$\mathbf{q}(k) = \mathbf{q}^{pred}(k) + \delta\mathbf{q}^{corr}(k),$$

followed by normalization to preserve the unit-norm constraint and ensure that the quaternion remains a valid rotation representation.

This additive update constitutes a first-order approximation that locally approximates the spherical domain as the tangent plane, and it is valid under the assumption of small correction increments, i.e., when the innovation remains sufficiently small. Although an error-state formulation would provide a more rigorous treatment of orientation corrections, the additive approach was adopted here for simplicity and during the testing phases in simulations did not exhibit noticeable drawbacks or stability issues.

Coupling considerations and theoretical limitations.
Although conceptually straightforward, this strategy introduces a subtle but important issue. The Kalman Filter framework assumes that measurements are independent from the predicted state estimate; however in this case, the NMPC velocity output is not an independent measurement: it is computed using the robot dynamic model that relies on the current estimated state as an input.

This creates a feedback coupling between the estimator and the controller. The EKF estimate is used by the NMPC, along with other information, to generate the control input and compute other quantities, including the predicted velocities, which are then reintroduced into the EKF as corrective measurements. Such circular dependence violate the independence assumptions underlying the Kalman filter framework and can potentially lead to overconfidence, inconsistency, or biased corrections. This effect could be mitigated by tuning the measurement noise matrix R to give more weight to the system model, however this strategy must be employed with caution.

3.5.2. MPC-Model-based EKF with IMU corrections

This second fusion strategy explores the complementary architecture in which the NMPC predictions are used as the main source of state propagation, while the IMU measurements are exploited for correction.

To clarify the structural difference between the two approaches, let us denote the estimation error as

$$\tilde{x}_k = x_k - \hat{x}_k, \tag{3.56}$$

where x_k is the true state and $\hat{x}_k$ is its estimate.

The Extended Kalman Filter is designed so that the propagated covariance P_k approximates the true error covariance $E[\tilde{x}_k \tilde{x}_k^T]$, under the assumption that the innovation term is statistically independent from the estimation error.

In the first strategy, the EKF state was propagated using inertial integration, and the NMPC velocity was injected as a measurement. Since the NMPC output depends on the estimated state fed back to the controller, the resulting measurement implicitly depends on $\hat{x}_k$, and therefore on $\tilde{x}_k$. As a consequence, the innovation term becomes statistically correlated with the estimation error. This violates the standard Kalman filter assumption that the innovation and the estimation error are uncorrelated, which is required for the optimal computation of the Kalman gain and for the consistency of the covariance update.

In the present formulation, the roles are reversed, and the dependence of the NMPC output on the estimated state affects the process model rather than the innovation term. The closed-loop dependence therefore appears as a modeling mismatch in the state prediction step, influencing the propagation of the estimation error and its covariance, while the innovation remains driven by independent and external sensor measurements.

It is important to emphasize that this formulation does not eliminate the coupling between controller and estimator, as the NMPC output still depends on the estimated pose. However, the structural location where this dependence enters the filter differs between the two strategies: in the first case it directly affects the innovation and therefore the Kalman gain computation, whereas in the second case it affects the process model and the error propagation. The purpose of this second experiment is therefore to investigate how this different structural placement influences the state estimation.

State.
The state remains the same as in the previous formulation:

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{p}(k) \\ \mathbf{v}(k) \\ \mathbf{q}(k) \end{bmatrix} \in \mathbb{R}^{10}, \tag{3.57}$$

where $\mathbf{p}(k) \in \mathbb{R}^3$ is the base position in the world frame, $\mathbf{v}(k) \in \mathbb{R}^3$ is the base linear velocity in the world frame, and $\mathbf{q}(k) = [q_w, q_x, q_y, q_z]^\top$ is the unit quaternion representing the base orientation.

System model (NMPC-based propagation)

In this formulation, the system model is built starting from the quantities easily obtainable from the robot controller; namely the base linear velocity $\mathbf{v}_{\text{mpc}}(k)$ and the base angular velocity $\boldsymbol{\omega}_{\text{mpc}}(k)$.

To be able to build a model from these 2 information, one possible way is to obtain an acceleration estimate by differentiating the NMPC velocity signal rotated in the world frame:

$$\mathbf{a}_{\text{mpc}}(k) \approx \frac{\mathbf{v}_{\text{mpc}}(k) - \mathbf{v}_{\text{mpc}}(k-1)}{\Delta t}. \tag{3.58}$$

And use this, along with the angular velocity $\boldsymbol{\omega}_{\text{mpc}}(k)$ as the control input for propagation.

It is important to note that numerical differentiation inherently amplifies and adds high-frequency noise. In a real implementation, this issue would require either explicit filtering or appropriate noise modeling within the EKF. In the present case, this effect is implicitly accounted for, by tuning the process noise matrix, without introducing additional modeling or filtering stages.

The discrete-time propagation equations therefore become very similar to the previous case:

$$\mathbf{q}(k+1) = \mathbf{q}(k) \otimes \delta\mathbf{q}(\boldsymbol{\omega}_{\text{mpc}}(k)\Delta t)\,, \tag{3.59}$$

$$\mathbf{a}_w(k) = \mathbf{R}(\mathbf{q}(k))\,\mathbf{a}_{\text{mpc}}(k), \tag{3.60}$$

$$\mathbf{v}(k+1) = \mathbf{v}(k) + \mathbf{a}_w(k)\,\Delta t, \tag{3.61}$$

$$\mathbf{p}(k+1) = \mathbf{p}(k) + \mathbf{v}(k)\Delta t + \tfrac{1}{2}\mathbf{a}_w(k)\Delta t^2. \tag{3.62}$$

Compared to IMU-based propagation, the gravity term does not appear explicitly in the equations since $\mathbf{a}_{\text{mpc}}(k)$ is computed starting from the NMPC output velocity, expressed in the body frame.

Linearization

The Jacobian of the system preserves the same block structure introduced in the IMU-based formulation, but with a simplified dependence on the state. Since $\mathbf{a}_{\text{mpc}}$ is treated as a direct control input and its dependence on the orientation is not explicitly modeled, the resulting Jacobian matrix becomes:

$$\mathbf{G}_k = \begin{bmatrix} \mathbf{I} & \Delta t\,\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{G}_{q,q} \end{bmatrix}, \tag{3.63}$$

where $\mathbf{G}_{q,q}$ is the same quaternion propagation term detailed in Paragraph 3.5.1. In contrast to the IMU-based propagation, the coupling terms linking orientation to position and velocity (i.e., $\mathbf{G}_{v,q}$ and $\mathbf{G}_{p,q}$) are absent. This is due to the fact that the dynamics are not explicitly driven by a rotated body-frame acceleration term $\mathbf{R}(\mathbf{q})\mathbf{a}_b$, but directly by $\mathbf{a}_w$; consequently, the cross-coupling blocks between position and orientation vanish in this simplified NMPC-driven model.

As a result, this aspect becomes important below, in the correction design. Since translational and rotational errors are less interconnected through the process model, two different measurement are required to properly constrain and correct all the states.

Observation model (IMU-based corrections)

The corrective information are provided in this architecture by an IMU. The absence of cross-coupling terms in the process Jacobian implies that rotational errors do not automatically propagate into translational states and vice versa. For this reason two measurements are needed here for the corrections to reach all the state variables:

- a base linear velocity estimate $\mathbf{v}_{\text{imu}}(k)$, obtained by integrating the IMU acceleration (with gravity removed using the orientation estimate of the IMU);
- an orientation reference $\mathbf{q}_{\text{imu}}(k)$, obtained by integrating gyroscope measurements.

The measurement vector is defined as:

$$\mathbf{z}(k) = \begin{bmatrix} \mathbf{v}_{\text{imu}}(k) \\ \mathbf{q}_{\text{imu}}(k) \end{bmatrix}, \qquad \mathbf{h}(\mathbf{x}(k)) = \begin{bmatrix} \mathbf{v}(k) \\ \mathbf{q}(k) \end{bmatrix}. \tag{3.64}$$

The corresponding measurement Jacobian is constant:

$$\mathbf{H} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{3.65}$$

The quantities used for correction are not direct measurement of the accelerometer or the gyroscope but they are IMU-derived quantities. An alternative, cleaner approach would consist in directly using the gyroscope measurement $\boldsymbol{\omega}$imu in the correction step, adding the angular velocity as a new estimated state in the EKF. In contrast, $\mathbf{q}_{\text{imu}}$ and $\mathbf{v}_{\text{imu}}$ are obtained through time integration of respectively angular velocity and acceleration, therefore, in a real-world scenario, they would be affected by integration drift. In the present study, however, the analysis is carried out in simulation using an ideal IMU model, so this drift-related issue does not arise. Nevertheless, in order to keep the formulation simple and to allow a clearer comparison between the two fusion strategies, the correction

was implemented using the two IMU-derived references.

It is worth to remember that, as shown in Paragraph 3.5.1, the additive corrective update performed by the EKF on the quaternion state is an approximation, valid under the assumption of small angle differences between the prediction and the correction.

Limitations and Drawbacks

In this configuration the closed-loop dependence manifests as a modeling approximation in the process dynamics rather than as a direct alteration of the innovation statistics. The practical impact of this structural choice depends on the tuning of the process and measurement covariance matrices Q and R, which regulate the relative confidence assigned to the NMPC-based propagation model and to the IMU-based correction.

One limitation of this approach lies in the numerical differentiation used to obtain the acceleration estimate in (3.58). Differentiation inherently amplifies high-frequency noise and discretization effects and in a real-world scenario it would require explicit filtering or a more refined modeling strategy to avoid injecting artificial dynamics into the propagation model.

Another important drawback concerns bias modeling. In the IMU-based propagation case, accelerometer and gyroscope biases can be naturally introduced as additional states, directly influencing the system dynamics and coupling with position, velocity, and orientation. In contrast, when the propagation model is based on NMPC outputs, the system equations are no longer driven directly by raw inertial measurements. As a consequence, introducing IMU bias states into this model becomes conceptually less straightforward, since the propagation dynamics do not explicitly depend on IMU signals. This reduces the flexibility of the estimator and complicates the extension toward a more realistic implementation.

4 | Results

This chapter presents and discusses the experimental results obtained using the state estimation approaches developed in the previous chapters. The main objective is to assess whether the theoretical considerations and modeling choices introduced in the methodology are reflected in coherent and meaningful experimental outcomes, and to evaluate the practical behaviour of the proposed estimators.

The results are organized progressively, starting from simple signal-level analyses and moving towards more structured state estimation experiments. The first part investigates the effect of basic signal pre-processing techniques applied to raw IMU accelerations, namely low-pass and high-pass filtering. Although these approaches are not part of the final estimation framework, they provide useful insights into the characteristics of inertial signals and help motivate the choices adopted later.

Then the focus shifts to the evaluation of an IMU-only state estimation approach based on an Extended Kalman Filter with zero-velocity updates. In this context, an external vision-based system is used to acquire ground-truth data, enabling a direct comparison between estimated and measured trajectories. The experiments are designed to analyze the achievable accuracy and temporal consistency of a dead-reckoning estimator relying solely on inertial measurements, with particular attention to the effect of the frequency of stationary phases during movements.

Finally, the chapter moves towards the analysis of two estimation strategies that aim at fusing IMU data with information provided by a quadruped robot controller, with the objective of estimating the robot pose. Although the integration with the NMPC is still under development, preliminary results are presented to explore how controller-level predictions can be exploited as an additional source of information for state estimation, highlighting both the potential benefits and the current limitations of this approach.

4.1. Effect of LPF and HPF pre-processing of IMU accelerations

The effect of simple signal pre-processing techniques applied to the raw accelerometer measurements has been investigated through experimental tests in this section. In particular, low-pass and high-pass filtering strategies were evaluated as possible means to improve the quality of velocity and position estimates obtained through direct integration of inertial data. Although these approaches were ultimately not adopted in the final estimation framework, they are reported here for completeness and to better motivate the design choices made.

4.1.1. Low-pass filtering of acceleration

Low-pass filtering of the accelerometer signal was first investigated with the aim of reducing high-frequency noise and obtaining a smoother signal prior to integration. The raw acceleration measurements were filtered using a low-pass filter, trying different cutoff frequencies, and the resulting acceleration signals were then double integrated to obtain velocity and position estimates.

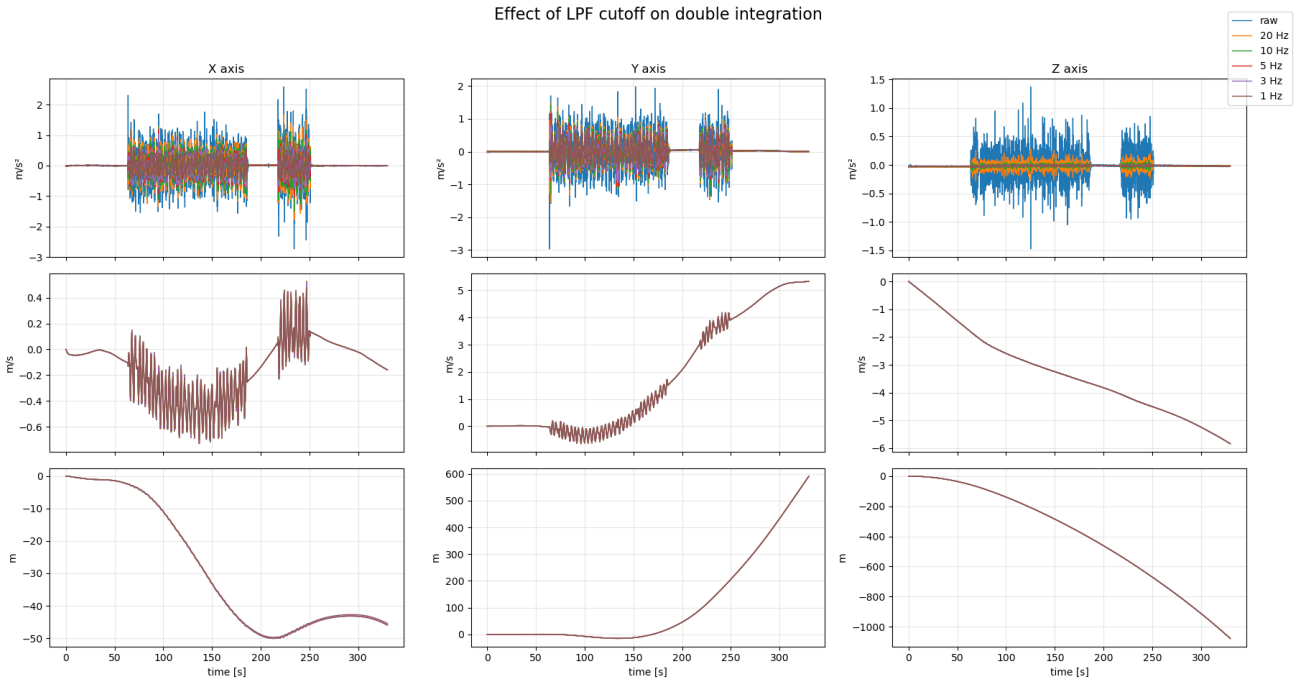

Figure 4.1: Effect of different LPF cutoff on double integrated acceleration

Figure 4.1 reports the results of this analysis, showing in 3 rows the acceleration, velocity,

and position along the three axes for different LPF cutoff frequencies, together with the unfiltered case. The position, as already stated, was computed via double integration, which corresponds to the propagation model used within the Kalman Filter during motion phases when no correction is applied.

From the plot is clear that despite significant differences in the filtered acceleration signals, the resulting position estimates remain almost unchanged across a wide range of cutoff frequencies. Noticeable differences appear only when using very low cutoff frequencies (e.g., 0.1 Hz), which however excessively attenuate the signal dynamics and therefore cannot be considered a viable choice. Similar behaviour was observed when varying the filter order, indicating that the overall outcome is largely insensitive to the specific LPF configuration.

This result can be explained by observing that time integration inherently acts as a low-pass filter. Since position is obtained through a double integration of acceleration, high-frequency noise components are already strongly attenuated by the integration process itself and as a consequence, the additional low-pass filtering of the acceleration signal has a negligible impact on the final position estimate. For this reason, LPF pre-processing of the acceleration signal was not found to provide benefits in this context.

4.1.2. High-pass filtering of acceleration

High-pass filtering of the acceleration signal was also considered, motivated by different considerations than those underlying LPF. In particular, HPF can be used to attenuate constant or slowly varying components of the signal, such as sensor bias or residual gravity contributions, which are known to be a major source of drift in inertial navigation.

Figure 4.2 illustrates the effect of high-pass filtering on the inertial signals and on the resulting integrated quantities. For each spatial axis, the figure reports the acceleration signals after HPF pre-processing, together with the corresponding velocity and position estimates obtained through double integration. Multiple curves are shown for each quantity, corresponding to different HPF cutoff frequencies applied to the acceleration measurements.

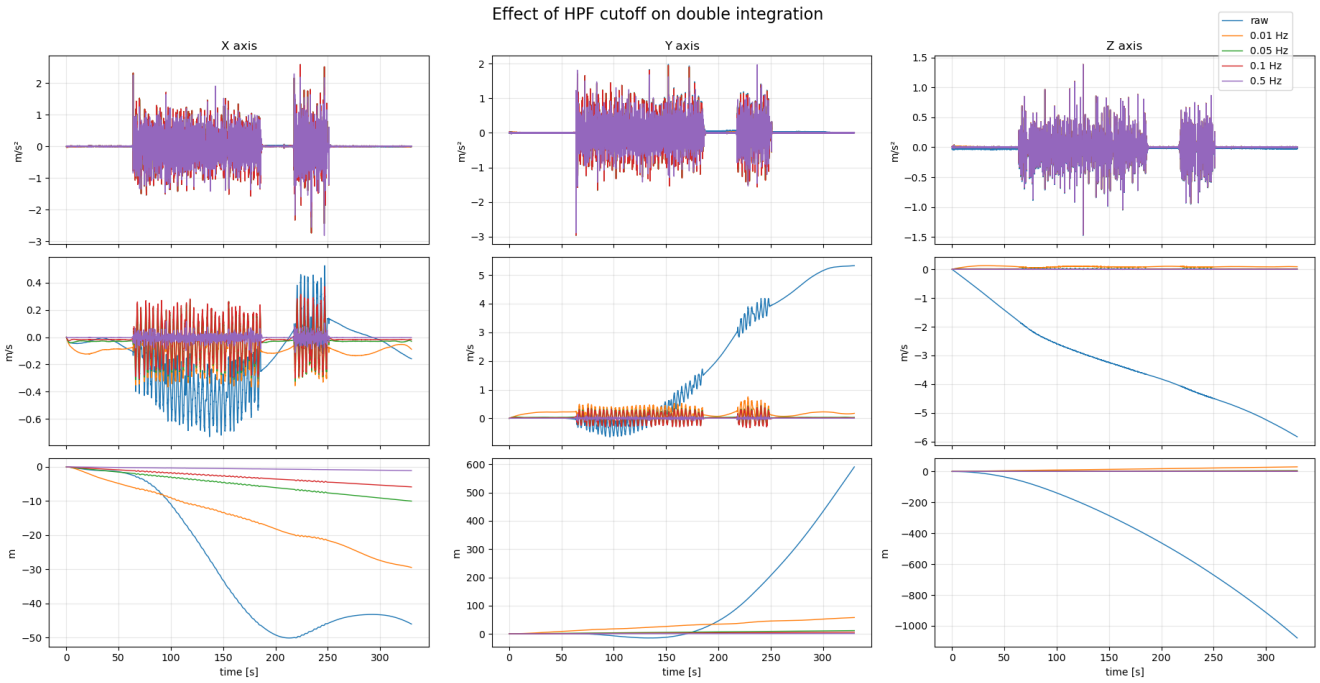

Figure 4.2: Effect of different HPF cutoff on double integrated acceleration

From the acceleration plots, it can be observed that decreasing the cutoff frequency progressively removes constant and slowly varying components, while preserving the higher-frequency dynamics of the signal. This behaviour directly propagates to the velocity and position estimates: as the cutoff frequency is lowered, the accumulated drift in velocity and especially in position is significantly reduced with respect to the unfiltered case. This effect is expected, since removing low-frequency components effectively suppresses the dominant contributors to integration drift.

However, despite these apparently promising results, this strategy is fundamentally incompatible with any strategy that uses double integration as the main tool to find an estimate of the position. High-pass filtering modifies the acceleration signal by removing low-frequency components that are physically meaningful and part of the true system dynamics. In a Kalman Filter context, where the process model relies on a physically consistent relationship between acceleration, velocity, and position it becomes more problematic. Applying an HPF alters the input to the process model in a way that is not explicitly represented within the filter equations, leading to an inconsistency between the assumed system dynamics and the actual propagated signal. Preliminary tests combining HPF pre-processing with the Kalman Filter confirmed this issue, resulting in degraded estimation performance compared to the unfiltered case.

It is worth mentioning that the right way to incorporate filtering strategies in the Kalman

filter framework is by explicitly modeling their dynamics and its logic in the model used by the Kalman filter. Obviously this would introduce additional complexity; consequently, without clear evidence of improved performance no explicit LPF or HPF pre-processing was applied to the accelerometer measurements in the final estimation framework.

4.2. Results of IMU-only state estimation

The objective here is to evaluate the performance of the proposed IMU-only based state estimation approach and to assess its consistency over time by comparing the estimated position against ground-truth data. In addition to validating the correctness of the estimation framework, the experimental results are also used to guide the tuning of key parameters of the estimator, including the process and measurement noise covariance matrices Q and R, as well as the parameters of the stationary detection.

4.2.1. Experimental setup and ground-truth acquisition

To obtain ground-truth position and orientation data, an external vision-based system was employed, consisting of a calibrated camera observing an ArUco marker rigidly attached on the IMU. By detecting the marker in the camera frame, it is possible to estimate the pose of the marker with respect to the camera, which is then used as a reference for evaluating the inertial state estimates.

To assess whether the vision-based system could be reliably used to acquire ground-truth position data, a simple preliminary experiment was performed. The IMU, with the ArUco marker rigidly attached, was placed on a planar surface and manually moved along a straight line between two reference points separated by 30 cm. The motion was guided laterally by a straight guide, ensuring that the displacement occurred predominantly along a single direction (the y-axis). The same motion was repeated multiple times, moving back and forth between the two reference positions, in order to qualitatively evaluate both the accuracy and the repeatability of the vision-based pose estimation. The measured trajectories, some of which are reported in Figure 4.3, represent the position of the marker expressed with respect to the reference frame detected on the marker at the first time instant.

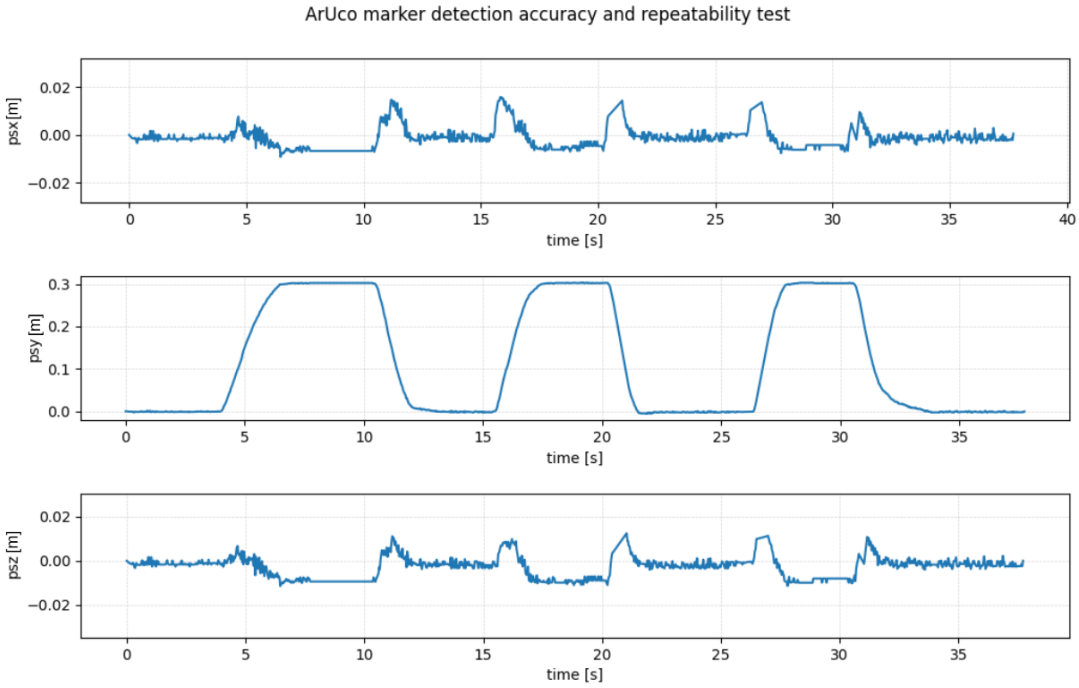

Figure 4.3: Position of the ArUco marker in space with respect to a reference frame located in its initial position

The resulting trajectories exhibited consistent displacement magnitudes and repeatable motion patterns across multiple trials, indicating that the system provides sufficiently stable and reliable position estimates for the purpose of ground-truth comparison.

While this approach provides accurate relative pose measurements of the marker with respect to the camera, it introduces the additional challenge of expressing the pose of the marker and the pose given by the IMU in the same reference frame, since a meaningful comparison between estimated and ground-truth trajectories requires all measurements to be expressed in a common reference frame. A schematic representation of the experimental setup and the involved reference frames is reported in Figure 4.4 for better understanding.

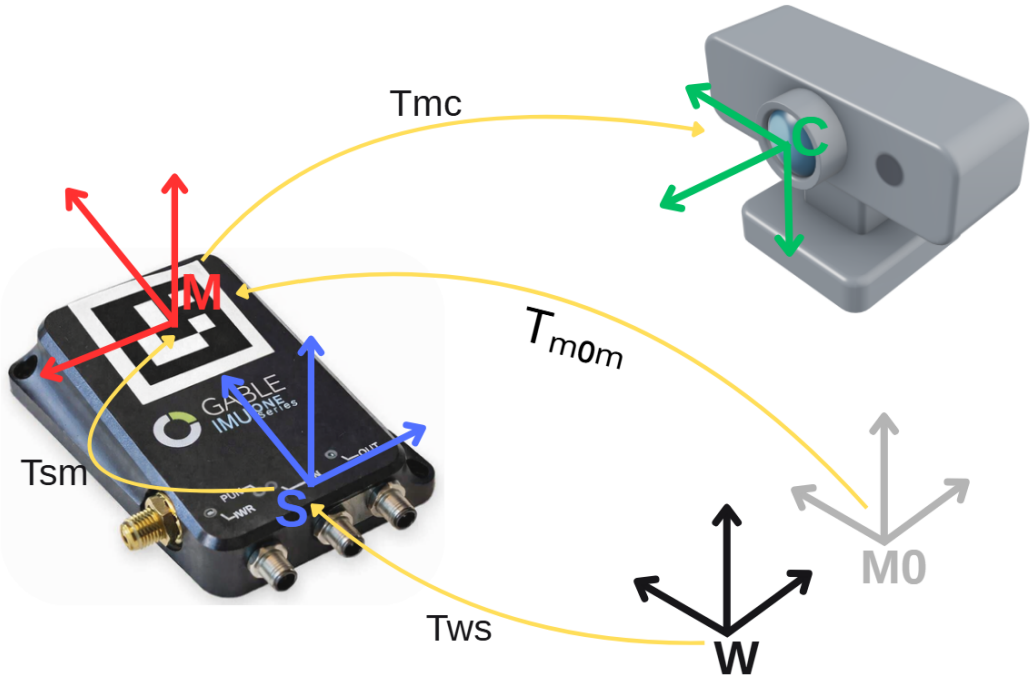

Figure 4.4: Schematic representation of the experimental setup illustrating the layout of all the different reference frames.

Before diving deeper into the methods adopted to align the different reference frames, it is useful to briefly refresh what are the involved coordinate systems and transformations, as well as clearing up which quantities are known and which must be estimated.

The Kalman Filter provides an estimate of the position expressed in a world reference frame W while the orientation is represented by the rotation R_{ws}, which describes the orientation of the sensor frame S with respect to the world frame.

On the vision side, the camera provides the pose of the ArUco marker frame M with respect to the camera frame C, expressed as a homogeneous transformation T_{mc} in which the translational part can be written as P_c.

To define a consistent reference for comparing IMU and marker positions these steps are performed: the first detected marker pose T_{m_0c} is stored and used as an initial reference. Subsequent marker positions are expressed with respect to this initial marker frame M_0: $\mathbf{p}_m = T_{m_0c}\,\mathbf{p}_c$; yielding marker trajectories that are independent of the absolute camera placement.

The remaining unknown is the rigid transformation T_{ms} relating the marker frame M to the sensor frame S. Once this transformation is known, marker positions expressed in the

initial marker frame M_0 can be mapped into the world frame according to

$$\mathbf{p}_w = T_{ws}\, T_{ms}\, \mathbf{p}_{m_0},$$

thus enabling a direct and meaningful comparison between the vision-based ground-truth data and the inertial state estimates.

Brief overview of the methods used for alignment of the two frames

In principle, this alignment problem was addressed as an **hand-to-eye calibration** procedure, aiming at estimating the rigid transformation between the camera and IMU frames. However, in the considered experimental setup, this approach proved to be unreliable. A primary reason is the presence of drift (always) in the IMU-based position estimates, which violates the assumptions underlying the hand-to-eye calibration formulation and leads to poor convergence of the calibration algorithm (we are trying to find a transformation to map the points acquired by the camera to the one acquired by the IMU, but on the IMU points there is a changing error, the drift, that create problems). Additional factors may have further degraded the calibration accuracy, including the very small magnitude of the translational offset to be estimated (in the order of a few centimetres) and possible small temporal misalignment between the camera and IMU data streams. As a result, the estimated transformation was found to be inconsistent and unsuitable for reliable trajectory comparison.

For these reasons, a simpler and more robust alignment strategy was adopted, at the cost of introducing a small error. The relative orientation between the IMU and the marker frame R_{ms} was **manually set** by estimating the fixed rotation matrix that best aligns the IMU sensor frame S, with the marker frame M, detected on top of the ArUco marker. To minimize the impact of drift in the inertial estimate, a single straight-line motion of short duration was acquired, approximately aligned with one principal axis of the IMU.

In a first step, a coarse alignment was performed by visually comparing the 2 plots of the 3 position axes and applying discrete rotations to bring the axes into consistent directions. Following this, a finer refinement was performed to accurately align the horizontal plane, since the vertical axis was already consistently oriented upwards in both reference frames. The horizontal displacement components of the ArUco-based trajectory were measured, and the corresponding rotation angle required to align the trajectory along a single axis (consistently with the IMU-based estimate) was computed using basic trigonometric relations. This angle was then converted into a rotation matrix and composed with the previously estimated one. The procedure was then repeated a small number of times

tuning the angle until a satisfactory overlap between the two trajectories was obtained. The comparison between before and after alignment of the two frames can be observed in Figure 4.5.

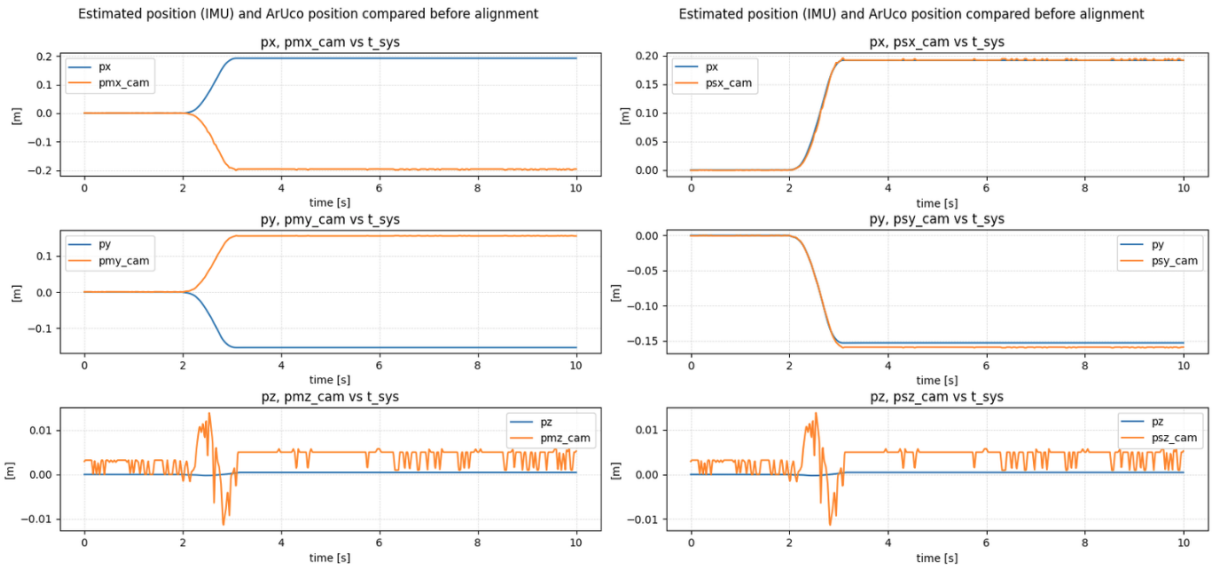

Figure 4.5: Before and after the alignment with the rotation matrix: marker position in yellow while IMU estimated position in blue.

It should be noted that this alignment method introduces a small residual error, mainly due to a slight translational offset between the physical origins of the marker and IMU frames. Nevertheless, this approximation is sufficient for the purpose of evaluating the temporal consistency and drift characteristics of the proposed estimation approach and for comparing different algorithms or tuning settings on the same dataset.

4.2.2. Experimental validation of the stationary detector

The stationary detector plays a key role in the IMU-only estimation framework by enabling zero-velocity updates (ZUPT) during motion pauses. The adopted detector is based on the evaluation of both accelerometer and gyroscope signals over a window of N samples, the detector classifies the system as stationary when both signals remain below predefined thresholds for a sufficient duration. When the system is detected to be stationary, the Extended Kalman Filter can enforce a zero-velocity pseudo-measurement, effectively stopping the growth of velocity drift and indirectly correcting other state components. For this reason, evaluating the correctness and robustness of the stationary detection mechanism is crucial for the overall performance of the IMU-only estimator.

A key challenge in stationary detection is avoiding false positive detections (i.e., classifying

the system as stationary while it is still moving). This error is particularly harmful, as it introduces incorrect zero-velocity constraints into the filter and can severely degrade the estimation. To mitigate this risk, conservative thresholds on acceleration and angular velocity were adopted and the window length N was tuned to reduce sensitivity to short-term fluctuations and measurement noise, at the cost of a small detection delay. This trade-off was considered acceptable, since delayed detections are generally less problematic than false stationary updates.

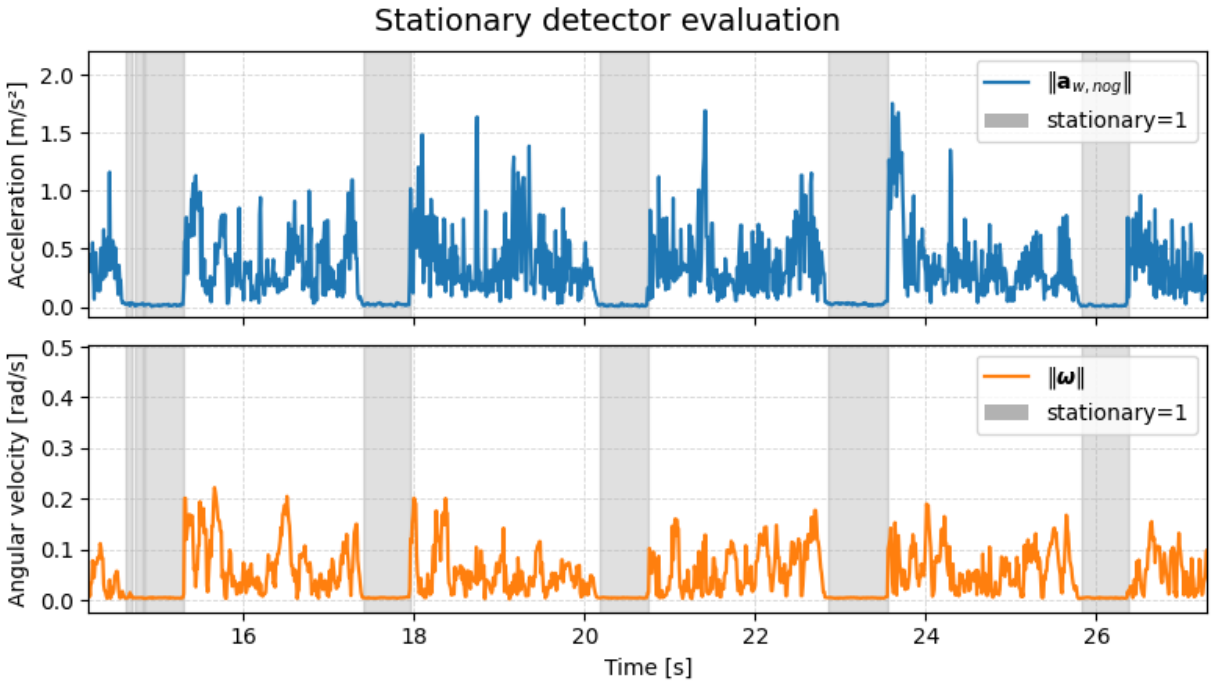

Figure 4.6: Norm of acceleration (above) and norm of angular velocity (below) with stationary detector in operation (gray bands)

Figure 4.6 shows the output of the stationary detector over time, together with the norm of the gravity-compensated acceleration and the norm of the angular velocity. The shaded regions indicate the time intervals classified as stationary by the detector.

The results are obtained from a dataset acquired using the physical IMU sensor, consisting of repeated planar motions mainly along the horizontal x–y directions, interleaved with stationary pauses. This type of motion sequence is representative of the conditions under which the ZUPT-based correction is expected to operate.

It can be observed that the detected stationary phases consistently correspond to intervals in which both the acceleration norm and the angular velocity norm remain low, confirming that the detector correctly identifies low-motion conditions suitable for zero-velocity

updates.

A limited amount of chattering can be observed only at the first transition between stationary and non-stationary phases on the left side of the plot, while the detector behaves consistently in the remaining intervals. This effect could be further reduced by introducing hysteresis thresholds or additional filtering on the signal used.

4.2.3. Analysis of estimated accelerometer bias

The estimation of an accelerometer bias term is a key component of the proposed EKF formulation, as it allows limiting drift in the estimated quantities by addressing its root cause. In the presented approach, the bias term is designed to absorb not only the slowly varying hardware-dependent offset, but also several additional error sources, including imperfections in gravity compensation, small orientation errors causing gravity leakage onto the horizontal axes, and other disturbances. Within the model, gravity and bias are treated as distinct quantities: gravity is removed using the orientation estimate, while the bias accounts for the residual effects that remain after gravity compensation.

The purpose of the following analysis is therefore to verify whether the estimated bias evolves at an appropriate rate and reacts consistently to the observed acceleration signals. To obtain the reported results, the process noise associated with the bias state in the Q matrix was tuned to allow the filter to estimate the bias at the appropriate rate, achieving a compromise between responsiveness and stability.

Figure 4.7 shows the estimated accelerometer bias components over time, superimposed on the corresponding gravity-compensated acceleration signals expressed in the sensor frame. The gravity-compensated acceleration $a_{s,\text{nog}}$ is reported instead of the raw acceleration to avoid confusion between gravity-related effects and bias estimation, since gravity is modelled as a constant and removed before in the filter.

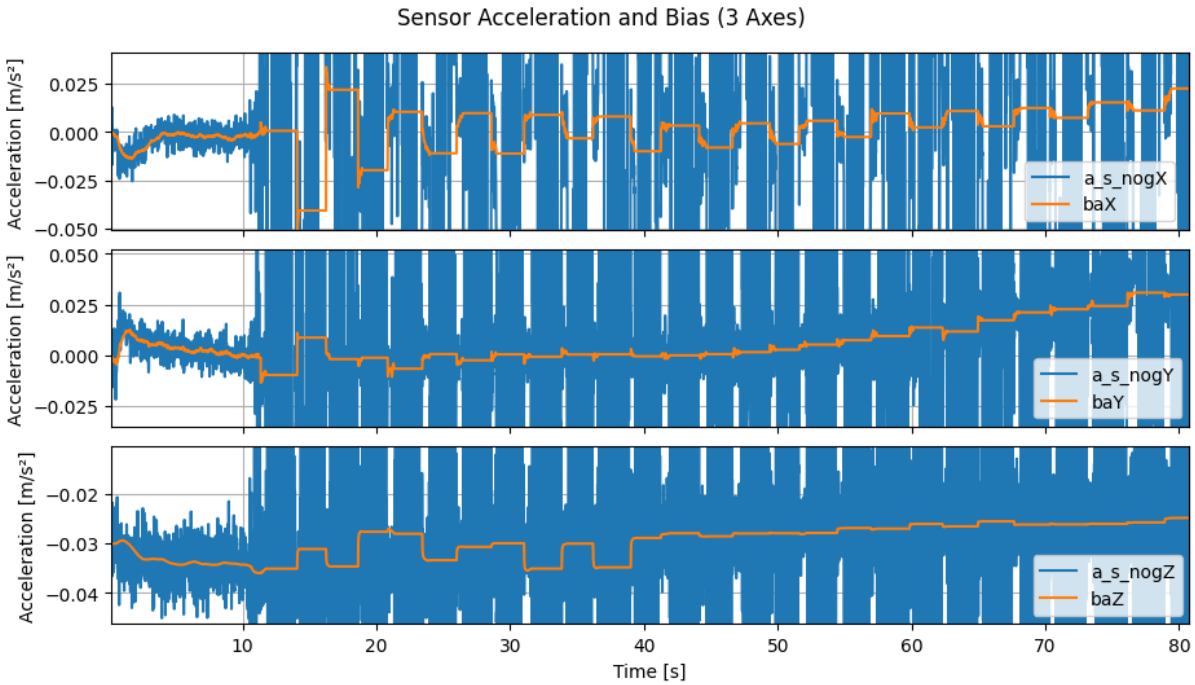

Figure 4.7: Estimated accelerometer bias components overlapped with gravity-compensated acceleration measurements in the sensor frame.

The results shown in Figure 4.7 are obtained from a dataset consisting of repeated planar motions, predominantly along the X-axis, interleaved with stationary pauses. As expected, the estimated bias remains approximately constant during motion phases, while it is mainly updated during stationary intervals, when zero-velocity updates provide the necessary information to correct accumulated errors. This behaviour is consistent with the assumption that the bias can be reasonably approximated as constant during moving phases, and that this approximation remains effective only when there are frequent pauses in the motion.

Two main effects can be observed in the estimated bias:

- First, a low-frequency variation is visible across the entire sequence, reflecting the expected behavior caused by hardware based bias and thermal effects.
- Second, a faster variation appears predominantly along the axis of motion (X), particularly at the transition between motion and stationary phases. These rapid changes correspond to bias updates applied immediately after motion ends, when the filter corrects velocity errors and redistributes part of the correction into the bias state.

Overall, the estimated bias follows the gravity-compensated acceleration signal with good

reactivity and consistency, indicating that the adopted modelling and tuning choices are effective. Although this approximation does not perfectly represent the true physical bias dynamics, it provides a practical and robust mechanism for compensating multiple sources of drift when frequent stationary phases are available.

4.2.4. Position estimation results under different motion patterns

The final set of results of this section evaluates the overall performance of the IMU-only estimation framework by directly comparing the estimated position with ground-truth trajectories acquired through the tracking of an ArUco marker rigidly attached to the IMU sensor. The objective is to assess the consistency of the estimated position over time and to highlight how the estimator behaves under different motion patterns and pause frequencies

Figure 4.8 shows the comparison between the estimated position and the ground-truth trajectory for a dataset composed of short motion segments lasting approximately 2-5 seconds, mainly along the X-axis.

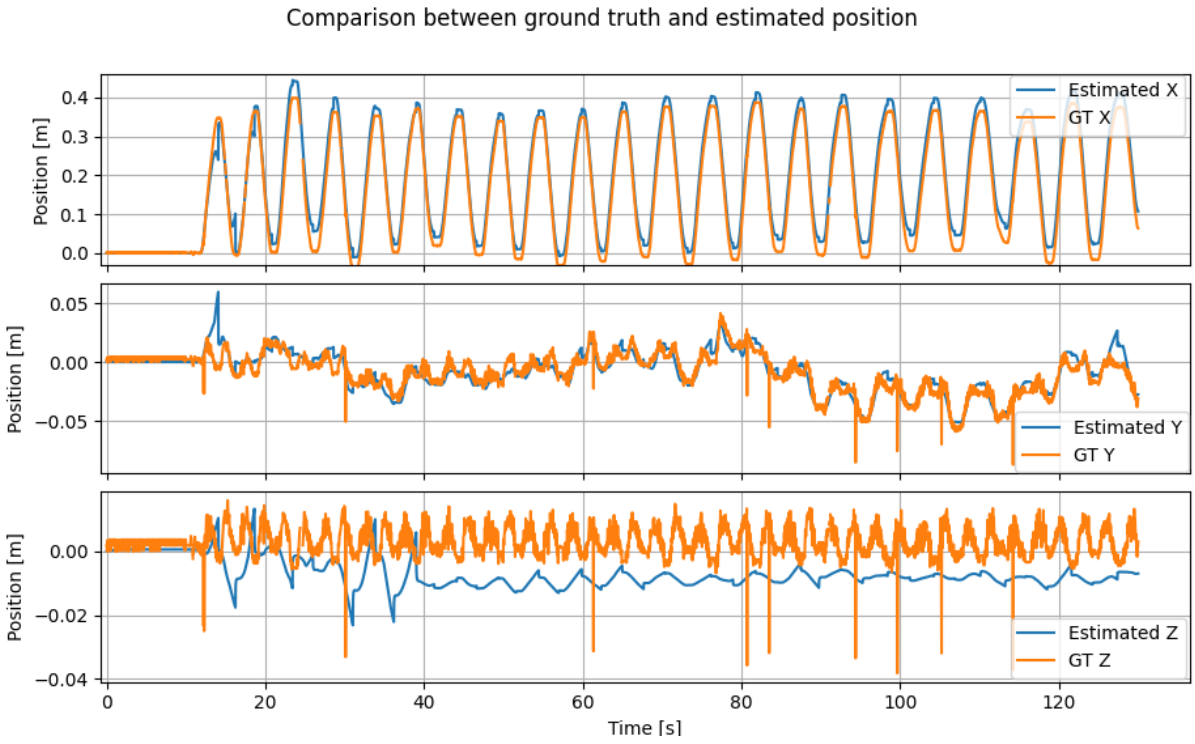

Figure 4.8: Comparison between estimated position and ground truth for short motion segments with frequent stationary phases.

In this scenario, the estimated position remains largely free from significant drift and preserves good accuracy over two consecutive minutes of acquisition. This behaviour is mainly due to the frequent corrections applied by the EKF through zero-velocity updates (ZUPT) during stationary phases, which effectively bound the accumulation of integration errors. By closely inspecting the trajectory, it is possible to identify the moments in which the corrections are applied, appearing as small discontinuities in the estimated position that compensate for the drift accumulated during the preceding motion phase. The main visible errors consist of a small constant offset of a few centimetres along the X-axis and approximately 1 cm along the Z-axis, which remain essentially constant throughout the experiment.

To test the estimator under more realistic and less constrained conditions, a second dataset was acquired in which the IMU was moved with random planar motions in the x–y plane, still interleaved with stationary pauses every 3-5 seconds or slightly longer. This motion pattern also avoids potential artifacts related to repeatedly moving the sensor back and forth along the same trajectory. Figure 4.9 shows the comparison between the estimated position and the ground truth for this dataset.

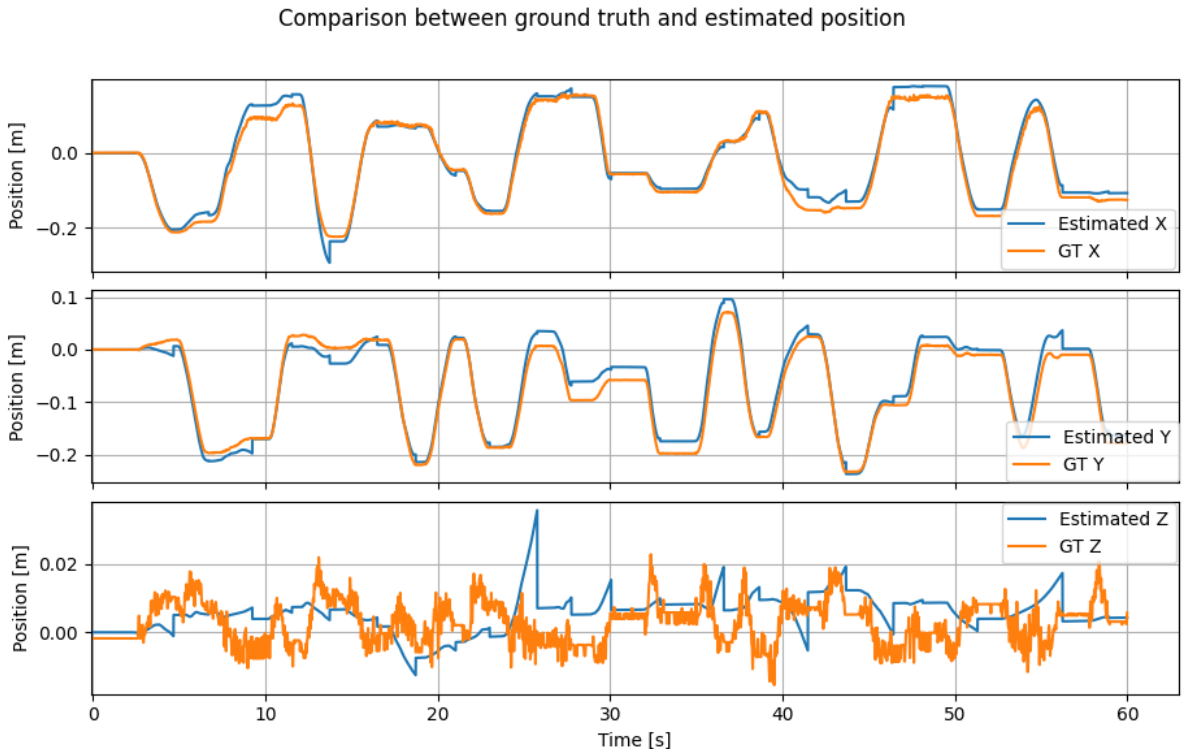

Figure 4.9: Comparison between estimated position and ground truth for random planar motions with frequent pauses.

Also in this case, the estimation remains accurate and well aligned with the reference trajectory. Only rare instances can be observed in which a correction slightly overshoots or is applied in the wrong direction. In general, however, the ZUPT-based corrections (more clearly visible here due to the longer motion segments) successfully bring the estimated position back towards the ground truth. Thanks to an improved tuning of the covariance matrices, no single axis exhibits significantly worse behavior than the others, resulting in a more robust and balanced estimate across all directions.

Finally, Figure 4.10 reports the estimation results for longer motion segments lasting approximately 10-20 seconds, with fewer stationary intervals and repeated forward and backward motions along the X-Y plane.

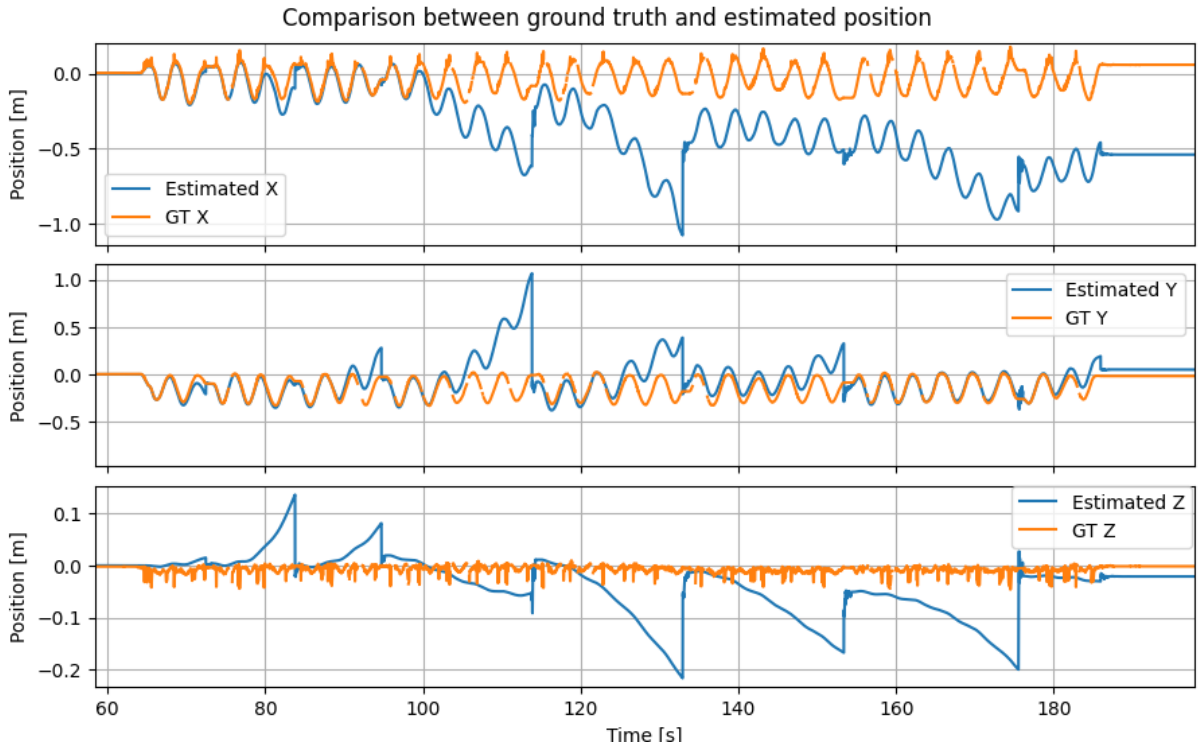

Figure 4.10: Comparison between estimated position and ground truth for longer motion segments with sparse stationary phases.

In this case, the position drift becomes significantly more evident between successive corrections. Moreover, the drift is not constant across motion segments, indicating that the acceleration bias estimated during the previous stationary phase evolves during the subsequent motion. While the estimation along the Y and Z axes remains reasonably accurate, the error along the X-axis grows substantially, reaching approximately half a meter over two minutes. Although the corrections still attempt to steer the estimate

back towards the correct position, they are no longer sufficient to fully compensate for the accumulated drift, leading to increasing estimation uncertainty and error after each correction.

This final scenario clearly highlights the intrinsic limitations of IMU-only state estimation relying exclusively on ZUPT corrections, whose effectiveness strongly depends on the frequency of stationary phases during motion.

While the presented IMU-only approach represents a useful baseline and an effective tool for understanding the behavior and limitations of inertial sensing, it is unsuitable for realistic quadruped locomotion where the hypothesis of frequent pauses cannot be respected. This motivates the transition towards a more robust state estimation framework that exploits additional information, in particular model-based predictions and control-related data provided by the robot controller.

4.3. Simulations results of different strategies to fuse IMU with NMPC data

The objective of this section is to evaluate whether the general idea of exploiting the robot controller as an additional source of information, in combination with inertial measurements, can represent a viable strategy for quadruped pose estimation, despite the potential coupling effects between controller predictions and the state estimator.

To this end, two alternative fusion strategies were explored. Both approaches rely on the same available resources (IMU measurements and NMPC outputs), but treating them within the estimation framework in different ways, resulting in two distinct estimator architectures:

- an IMU-based propagation model with NMPC-based velocity corrections,
- an NMPC-based propagation model with IMU-based corrections.

The purpose of these experiments is not to provide a final, fully optimized solution, but rather to qualitatively compare the two architectures, analyse their behaviour, and identify potential strengths and weaknesses. Particular attention is given to understanding whether one formulation exhibits more stable behaviour, reduced drift, or less sensitivity to coupling effects.

Simulation setup

All presented experiments were conducted in simulation using the MuJoCo physics en-

gine, given that the quadruped robot controller is still under development in the same environment (image of the simulation environment reported in Figure 4.11).

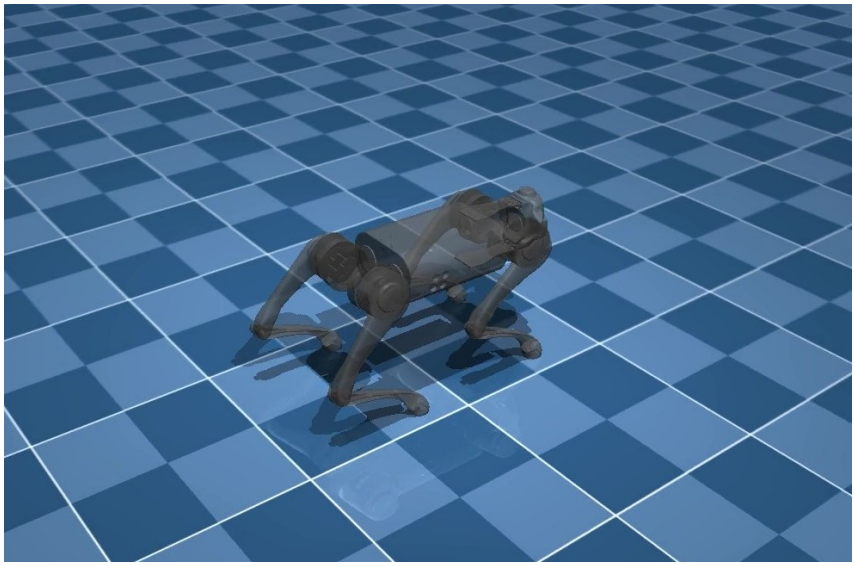

Figure 4.11: MuJoCo simulation running the quadruped controller and state estimator

The quadruped robot is controlled through the same Nonlinear Model Predictive Controller (NMPC) + Whole-Body Controller (WBC) architecture described previously, including trajectory interpolation and contact handling. The estimator was integrated within this simulation pipeline, allowing direct access to the controller outputs and simulated sensor signals.

Before integrating the proposed state estimators, the controller was operating using ground-truth state information as inputs, directly provided by MuJoCo. In the experiments presented here, the estimator output was in most cases fed back to the NMPC, resulting in a closed-loop configuration. Only in a few cases the controller continued to use ground-truth state information, as explicitly specified in the corresponding results.

A simulated IMU model was used and placed at the same position as the robot base frame. In this way, accelerations and angular velocities correspond directly to the base motion without additional offsets or mounting transformations. Sensor bias was not included in these tests, while Gaussian noise was added in some experiments in order to test the robustness of the estimate.

The simulated environment consists of a flat terrain scenario in which the quadruped NMPC controller is fed with a straight trajectory to be followed, along the X axis. This setup allows for controlled evaluation of the estimator behavior under repeatable and well defined motion conditions.

4.3.1. Results of IMU-model based state estimator

This section presents the simulation results obtained with the first EKF architecture: the IMU-based propagation model combined with NMPC predicted velocity corrections. The goal is to assess how effectively the NMPC velocity feedback can constrain inertial drift and whether the estimator, together with the controller, remain stable when the pose estimate is fed back into the input of the NMPC controller.

Model propagation alone

Figure 4.12 shows the position estimate obtained by pure model propagation, without any correction step and without feeding the estimated state back to the NMPC (open-loop configuration). This experiment was performed as a preliminary validation of the propagation model, isolating its behavior from correction mechanisms and controller feedback effects.

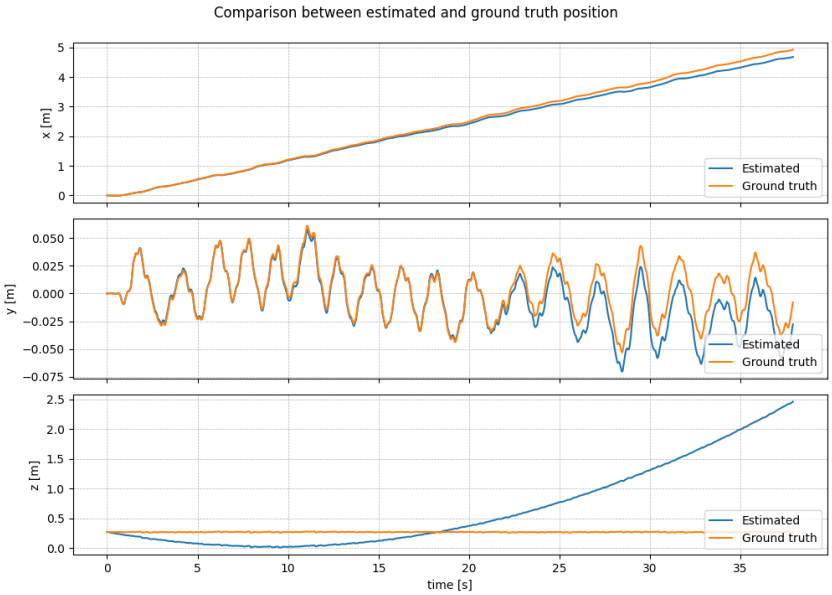

Figure 4.12: Estimated position and ground truth. Just model propagation

Since an ideal IMU is used in simulation (no bias and no measurement noise), the estimated position closely follows the ground-truth trajectory along the horizontal axes, confirming the correctness of the implementation of the propagation equations and of the frame transformations involved in the model. However, a noticeable drift can be observed along the Z axis, behavior that will also be present across other tests.

The vertical drift in position is likely related to the contact dynamics of the simulated quadruped. Even in the absence of sensor noise and bias, ground contact interactions generate high-frequency vertical accelerations that, once integrated, may lead to small

residual position drift. Additionally, minor inconsistencies between the simulator initial conditions and the estimator initialization, due to initial spikes of the contact forces, can accumulate over time in the absence of corrective updates. For these reasons in the next plots mostly just the X and Y axis will be analyzed.

Evaluation of EKF-estimation fed back into NMPC

In this experiment, the full IMU-model-based EKF with NMPC velocity corrections was tested in closed-loop configuration, meaning that the estimated pose was fed back to the NMPC as the current state input. Here the estimator actively influences the controller, creating a feedback interaction between state estimation and control.

To avoid instability and divergence, the measurement covariance R associated with the MPC-based velocity correction was tuned conservatively, assigning a relatively low weight to the NMPC velocity. Higher correction gains were observed to induce rapid divergence of the estimate, highlighting the sensitivity of this architecture to the coupling between controller outputs and estimator corrections.

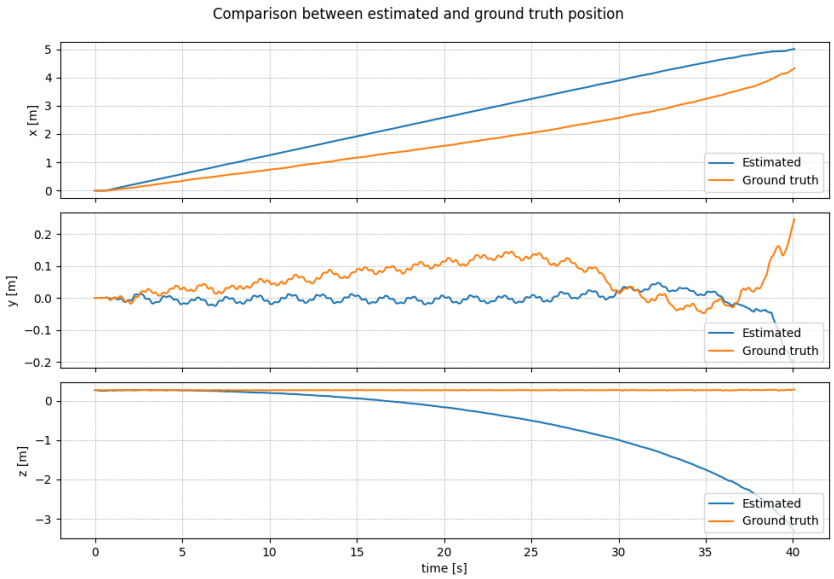

Figure 4.13: Estimated position and ground truth. Model propagation and correction, feedback of the estimate back into the controller

Figure 4.13 shows the estimated position compared with ground truth. It is expected that in this ideal case the inclusion of MPC-based velocity corrections does not improve the estimation performance. In several segments, the position estimate deviates more significantly from ground truth than in the pure IMU propagation case.

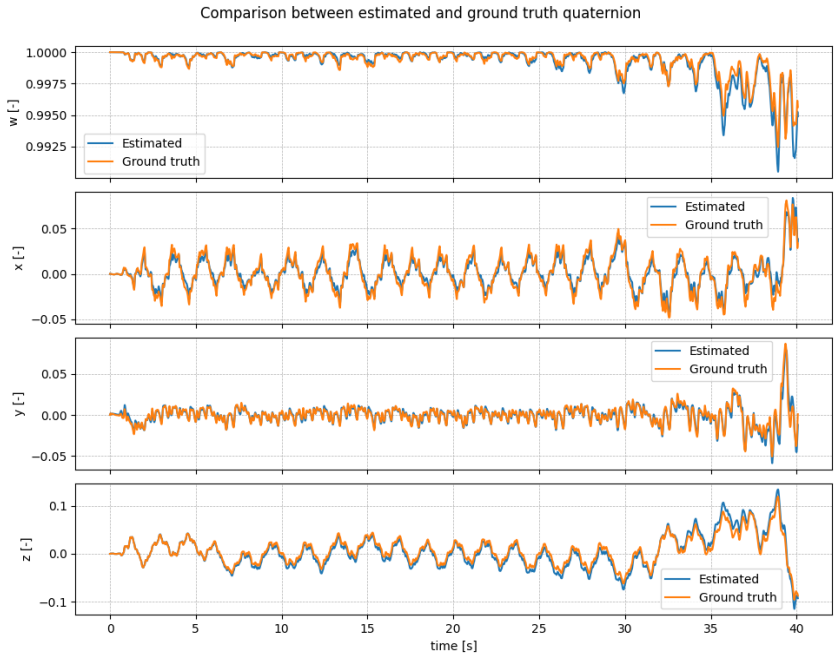

Figure 4.14: Quaternion components of the estimated orientation compared with ground truth.

The corresponding quaternion evolution, reported in Figure 4.14, is much more consistent in time and follows precisely the ground truth, mainly because affected only indirectly by the MPC correction.

To better understand the origin of this behavior of the position estimate, Figure 4.15 compares the NMPC-predicted velocity used as correction input with the ground-truth velocity. It can be observed that the NMPC velocity does not perfectly match the true base velocity, exhibiting a constant offset in some axis while a drift in others, and overall small inconsistencies over time.

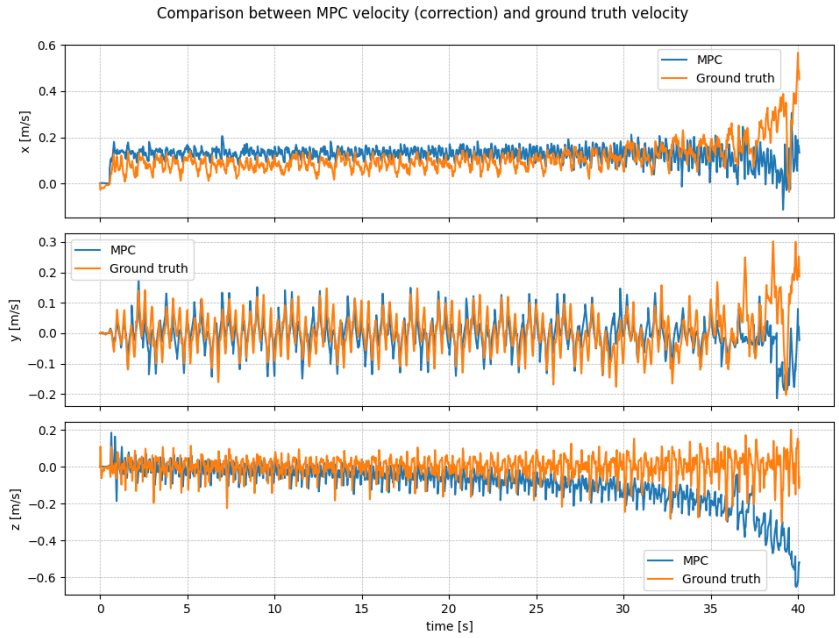

Figure 4.15: Comparison between NMPC-predicted base velocity (used for correction) and ground-truth velocity.

These results suggest that, in this configuration, the NMPC velocity does not provide a reliable correction signal and, when assigned excessive confidence, it introduces additional estimation errors due to the intrinsic coupling between controller predictions and the estimator state.

It is worth noting that the vertical drift observed along the Z axis could in principle be mitigated by explicitly modeling accelerometer and gyroscope bias states within the EKF and allowing them to be updated through the correction step. In a real-world scenario, similar drift phenomena would also affect the horizontal axes due to sensor biases and low-frequency disturbances, and would necessarily require explicit bias modeling and compensation. For this reason, the Z-axis drift is not considered the main limiting factor of this architecture; rather, the dominant issue lies in the structural coupling between controller-generated predictions and the estimator corrections.

4.3.2. Results of NMPC-model based state estimator

In this second fusion architecture, the propagation and correction roles are reversed with respect to the previous one. The NMPC outputs (base linear and angular velocity) are used to propagate the state estimate, embedding the controller dynamic model inside the EKF process model. The IMU instead, is exploited in the correction step through IMU-

derived quantities, namely an orientation reference (computed from the angular velocity measurement) and a velocity estimate (obtained integrating the acceleration measurement). Here, the corrective information is provided by the IMU, which is independent from the controller predictions, while the NMPC output acts as a model-based motion generator for the estimator.

The following results aim to evaluate whether this inversion of roles in the architecture improves estimator robustness and reduces the problematic feedback loop observed in the IMU-propagation architecture.

Propagation of the EKF model alone

Figure 4.15 shows the position estimate obtained by propagating the EKF using only the NMPC-based process model, without applying any correction step and without feeding the estimated state back into the controller.

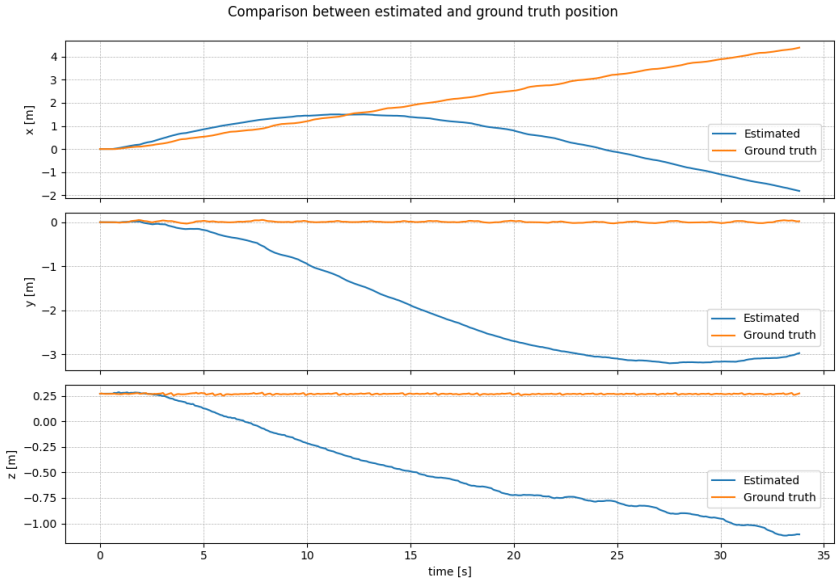

Figure 4.16: Comparison between estimated position and ground truth; just propagation of the NMPC-based model.

In contrast to the IMU-based propagation case in Figure 4.12, where the estimate decently followed the ground truth under ideal IMU assumptions, the NMPC-driven propagation exhibits significant divergence and instability. The position estimate departs rapidly from the reference trajectory, showing inconsistent behavior even in the absence of estimator-controller feedback.

This result represents a first indication that the motion information reconstructed from the NMPC outputs (through velocity differentiation and integration) probably does not

provide a sufficiently reliable propagation model.

Evaluation of the full estimator: feed-back of the estimate into the controller

After testing just the model propagation to have a baseline, we evaluate the full EKF architecture. In this configuration, two IMU-derived quantities (velocity and orientation) are used for correction, and the complete estimated state (position, velocity and orientation) is injected into the NMPC as the current state input. The estimator and controller therefore operate in closed loop, directly influencing each other.

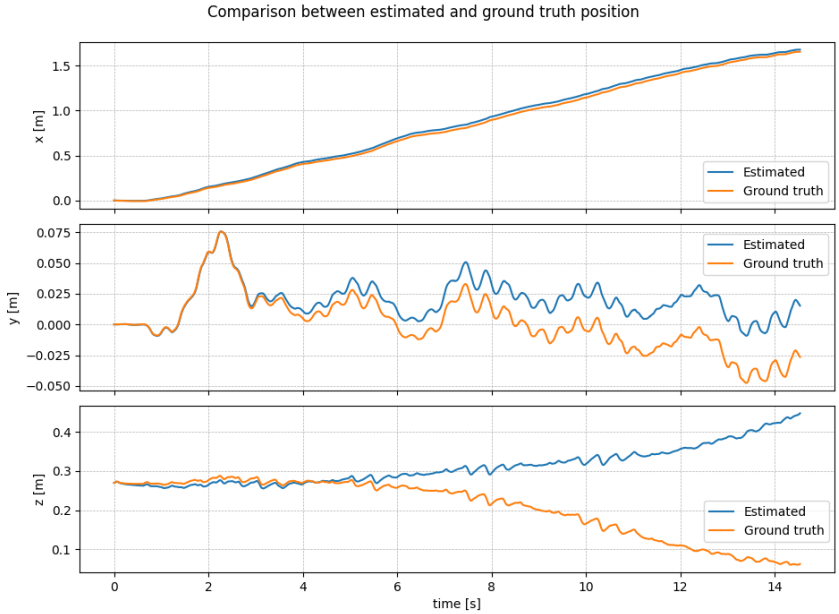

Figure 4.17: Estimated position and ground truth using NMPC-based propagation with IMU corrections and full state feedback into the controller.

From Figure 4.17, it can be observed that the behavior along the X and Y axes significantly improves compared to the pure NMPC-based propagation. The effect of the IMU correction is clearly visible: although the estimate is not perfectly accurate, it no longer diverges and appears considerably more stable over time. The closed-loop system does not exhibit the strong instability previously observed.

The Z axis represents a partial exception. The estimated vertical position drifts in the wrong direction, despite the robot operating on a planar terrain. This behavior is consistent with the issue already discussed in the previous chapter (see Section 4.3.1), where vertical drift was attributed to contact dynamics and to initial transient effects in the simulation. The vertical velocity obtained from IMU acceleration integration exhibits

drift (see Figure 4.18), and when such a drifting signal is used as corrective input, the resulting position estimate is inevitably degraded.

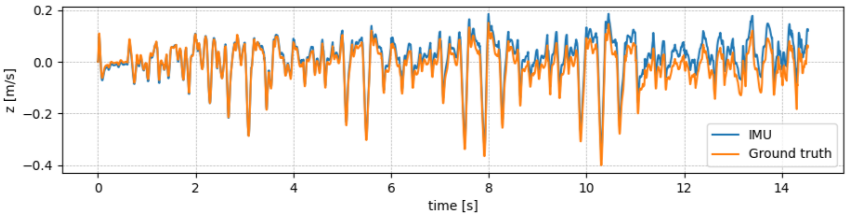

Figure 4.18: Z-axis of the velocity computed from IMU measurements and used for correction compared with ground truth.

Beyond the central focus of the analysis, the Z axis in Figure 4.17 also reveals an interesting effect worth noting. While the estimated position drifts upward, the ground-truth vertical position moves downward, in the opposite direction. This behavior highlights the closed-loop interaction between estimator and controller. The NMPC, receiving a state estimate indicating that the robot base is rising, attempts to maintain the reference height by generating control inputs that lower the robot. As a result, the true base height decreases, demonstrating that the controller is directly influenced by the estimated state.

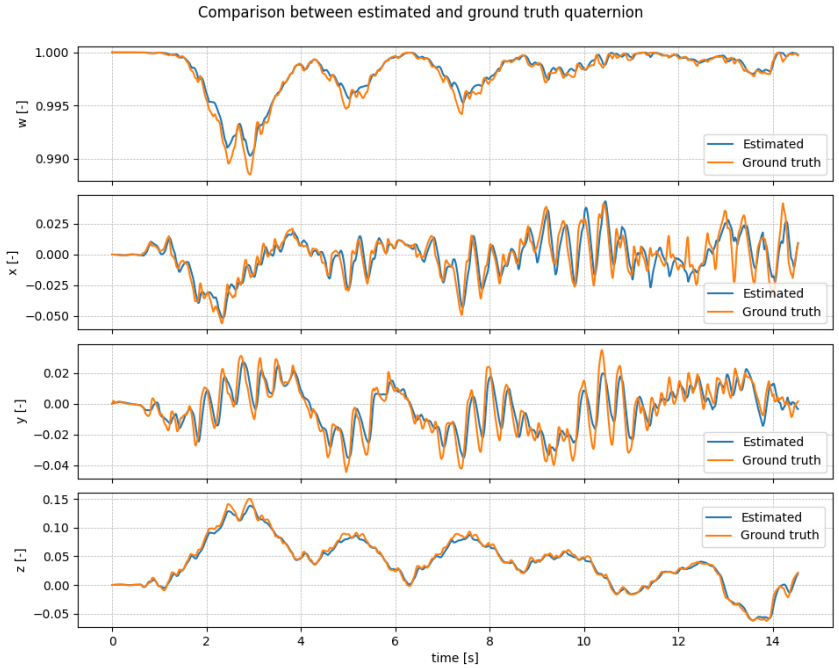

Figure 4.19: Quaternion components of the estimated orientation compared with ground truth in closed-loop configuration.

Figure 4.19 shows the quaternion components of the estimated orientation. In contrast to the vertical position behaviour, the orientation estimate remains stable and closely

follows the ground truth, without exhibiting significant drift or instability. The higher accuracy, with respect to the position estimate, is mainly due to the corrective signal used. The quaternion computed from the gyroscope of the IMU and used for correction, does not suffer from the contact-related effects affecting the accelerometer, and the corrective signal is therefore more accurate.

Diagnostic test on vertical correction

To further investigate the estimate inaccuracy, particularely observed along the Z axis, an additional test was performed. In the previous configuration, the robot was progressively lowering its base until the simulation became unstable, eventually leading to ground contact and failure. To verify whether the vertical corrective signal was the main source of the problem, the vertical component of the IMU-derived corrective velocity was artificially forced to the ground-truth value, while keeping the rest of the estimator unchanged and injecting the full state estimate into the NMPC.

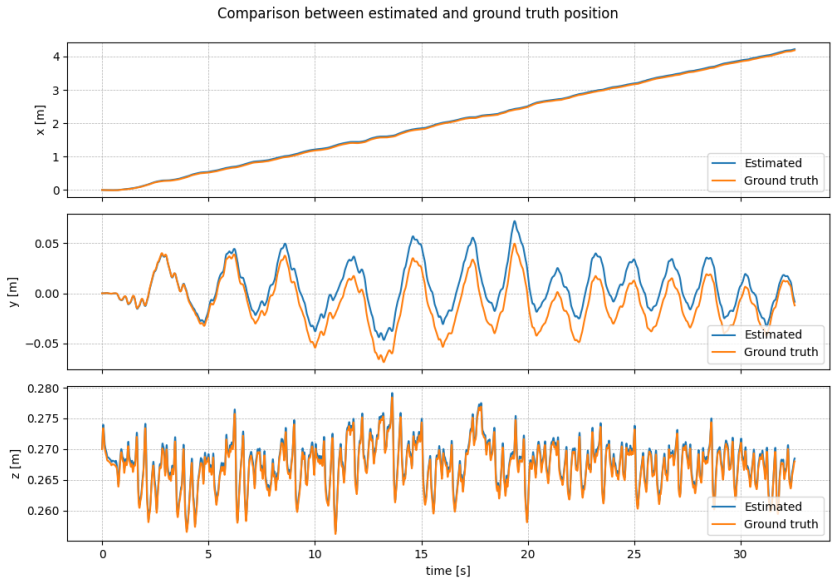

Figure 4.20: Estimated position with vertical corrective velocity forced to ground truth.

As shown in Figure 4.20, the overall behaviour improves noticeably not only in the Z component (as expected), but also in the X and Y components, confirming that the vertical correction was playing a dominant role in the observed issues. (The orientation behavior is not reported here, as no significant difference was observed with respect to the previous configuration.)

State estimate evaluation adding Gaussian noise

After validating the estimator behavior in ideal conditions, Gaussian noise was introduced to evaluate robustness under more realistic conditions. This step is necessary to assess whether the observed stability properties persist when measurement uncertainty is taken into account.

The same estimator configuration as in the previous test was retained: NMPC-based propagation, IMU correction with $v_{imu,z}$ forced to ground truth to avoid Z-drift, and full state feedback to the controller. Zero-mean Gaussian noise was added to IMU measurements and to encoder measurements fed to the controller. In particular, the standard deviations adopted for the Gaussian noise were computed based on the noise density values specified in the sensor datasheets. For the accelerometer, the datasheet specifies a noise density of

$$S_a = 70\ \mu g/\sqrt{\text{Hz}} = 70 \times 10^{-6} \cdot 9.81 \approx 6.87 \times 10^{-4}\ \text{m/s}^2/\sqrt{\text{Hz}}.$$

where S_a represents the one-sided amplitude spectral density of the accelerometer noise.

To obtain an equivalent discrete-time standard deviation σ_{acc} (in m/s^2) for sampled measurements, assuming a white-noise model, the variance over a bandwidth B is obtained from the power spectral density (PSD) as

$$\sigma_{acc}^2 = \int_0^B S_a^2\, df = S_a^2 B,$$

which leads to

$$\sigma_{acc} = S_a\sqrt{B}.$$

Assuming that the effective bandwidth is limited by the Nyquist frequency, the maximum noise band can be approximated as $B = \frac{f_s}{2}$, where f_s is the sampling frequency. Using $f_s = 200$ Hz:

$$\sigma_{acc} \approx S_a\sqrt{\frac{f_s}{2}} = 6.87 \times 10^{-4}\sqrt{100} = 6.87 \times 10^{-3} \approx 0.006\ \text{m/s}^2.$$

Similarly, for the gyroscope, starting from a noise density of $0.003\ ^\circ/\text{s}/\sqrt{\text{Hz}}$ reported in the datasheet, and applying the same procedure used for the accelerometer, the equivalent discrete-time standard deviation becomes:

$$\sigma_{gyro} \approx 5.2 \times 10^{-4} \approx 0.0005\ \text{rad/s}.$$

For the joint encoders, the noise levels were selected according to typical values adopted

in high-resolution robotic applications, ensuring realistic but non-dominant measurement uncertainty in the control loop, resulting in $\sigma_q = 0.0005$ rad and $\sigma_{\dot{q}} = 0.001$ rad/s.

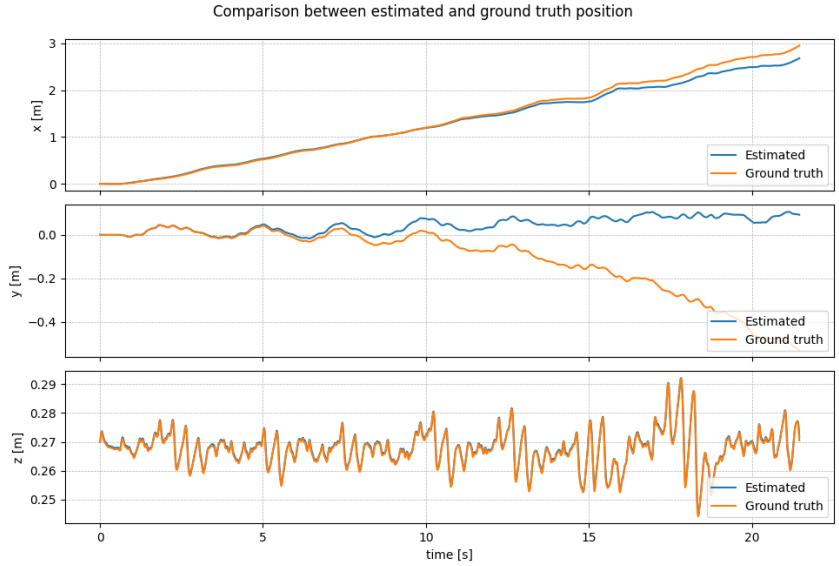

Figure 4.21: Evaluation of estimated position when adding Gaussian noise on IMU and encoder measurements.

As shown in Figure 4.21, the position estimate deviates more from the ground truth, compared to case without noise and a visible drift appears along the Y axis.

This increased position error can be directly attributed to the corrective velocity obtained from the IMU. As visible in Figure 4.22, the IMU-derived velocity exhibits a slightly increased drift with respect to the ground truth due to the injected measurement noise. Since this velocity is used in the correction step, its reduced accuracy propagates to the position estimate, leading to a less precise correction. The coherence between the drift direction observed in the velocity signal and the corresponding deviation in the position estimate further confirms this causal relationship.

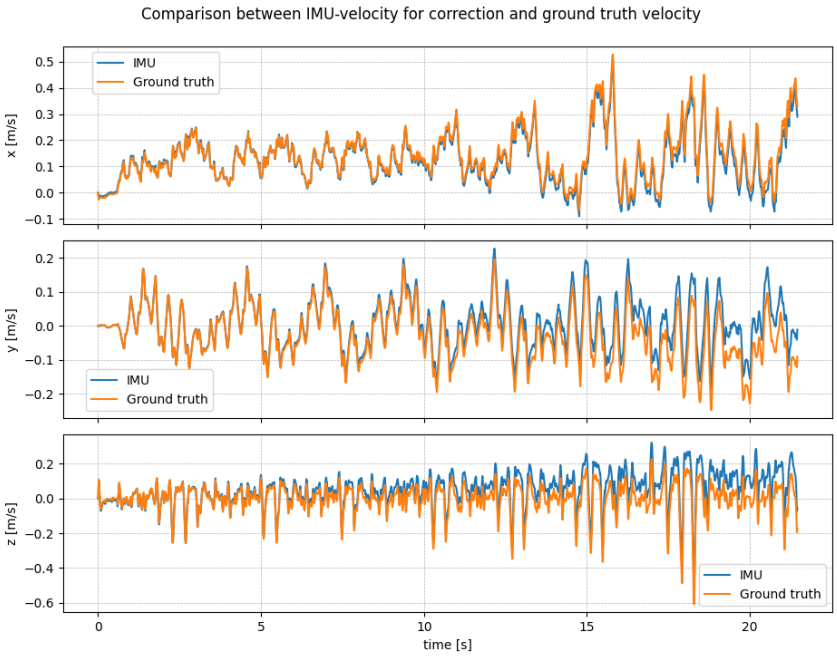

Figure 4.22: IMU-derived velocity used for correction under Gaussian noise.

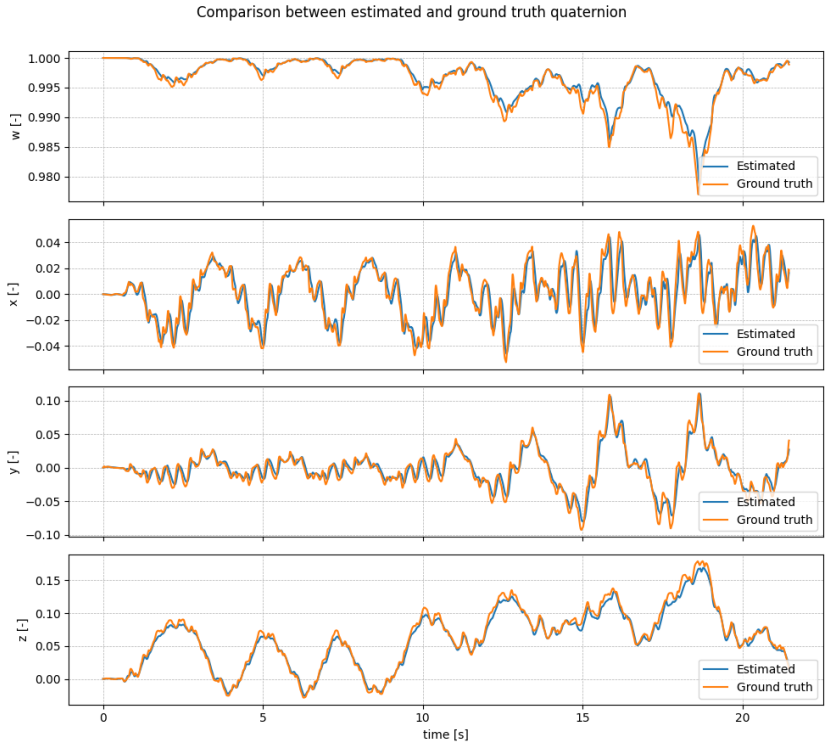

Figure 4.23: Quaternion components of the estimated orientation with noisy measurements.

In contrast, the orientation estimate shown in Figure 4.23 remains accurate, with only

slightly larger deviations from the ground truth compared to the noise-free scenario. No significant instability or drift is observed in the quaternion components.

Overall, bearing in mind that the vertical component was, also in this test, forced to ground truth, the estimation performance slightly degrades, but not significantly, when Gaussian noise is introduced. The comparison between the noise-free and noisy scenarios suggests that noise is not the primary limiting factor of this estimation strategy. Rather, other aspects appear to have a stronger impact on performance, such as the tuning of the covariance matrices and the use of integrated IMU-derived signals within the correction step. Since the corrective velocity is obtained through integration from IMU quantities, even small biases can accumulate over time, highlighting the necessity of explicitly modeling IMU biases within the estimator in order to prevent long-term drift of the corrective signal and of the estimate itself. However, in the present architecture, explicitly modeling IMU biases is considerably more challenging in practice. As discussed in Paragraph 3.5.2, the estimator process model is driven by the NMPC outputs rather than by a pure inertial model, meaning that the sensor dynamics are not directly embedded in the state formulation. Introducing bias states associated with the IMU would therefore be less straightforward and require extending a model that is not intrinsically sensor-based.

4.3.3. Discussion and comparison of the two fusion strategies

The simulation tests presented above allows drawing a number of qualitative conclusions regarding the two explored fusion architectures.

A first key outcome is that the **coupling between controller predictions and state estimation** represents a non-negligible limiting factor in both formulations. In the IMU-propagation strategy, the NMPC-predicted velocity is injected as a measurement to correct an estimate that is also fed back to the controller, creating a closed-loop interaction in which the correction signal is not independent from the estimated state. In practice, this leads to high sensitivity to the tuning of the measurement covariance causing instability and even divergence of the estimate.

The NMPC-based propagation strategy does not appear to mitigate the coupling observed in the first method, even if the dependence enters the filter in a different location of the filter. Overall, the experiments suggest that, in the considered setup, the **IMU information dominates the estimator behavior**, while the NMPC outputs can be used at most as weak additional constraints. Assigning higher confidence to controller-derived quantities (either as measurements in the first method or as propagation in the second)

tends to degrade performance, confirming that their current accuracy and consistency are not adequate to serve as a primary estimation source.

Another important aspect emerging from both approaches is the **use of derived, non-direct signals**. Depending on the formulation, either the propagation or the correction relies on quantities obtained through numerical differentiation (e.g., acceleration from $\mathbf{v}_{\text{mpc}}$) or integration (e.g., $\mathbf{v}_{\text{imu}}$ from IMU acceleration, or orientation from angular rate integration). While this can be acceptable in short simulations with ideal sensors, in real deployments these operations amplify noise and accumulate bias, and would require explicit bias modeling and careful filtering. This is particularly critical for the NMPC-propagation architecture, where the process model is not intrinsically sensor-based, making the inclusion of IMU bias states less straightforward.

In conclusion, these results indicate that controller outputs alone are not suitable as a correction source, and that they must be strongly supported by other corrections. The analysis supports the expectation that more established legged-robot state estimation approaches, based on contact information, leg kinematics, and leg odometry, are necessary to achieve long-term robustness in realistic locomotion scenarios. The two fusion strategies investigated here should therefore be interpreted as exploratory baselines, useful to highlight limitations of NMPC-based signals and to motivate future work toward contact-based estimation frameworks.

5 | Conclusions and future developments

The objective of this thesis was to investigate alternative strategies for quadruped state estimation, with particular focus on exploring whether the output of the robot controller (NMPC-based architecture) could be exploited as an additional information source within a state estimation framework. The underlying motivation stems from the necessity of bounding and supporting pure inertial pose estimation (IMU-based), that inevitably suffer from drift. However, in quadruped state estimation, this issue is typically mitigated by relying on exteroceptive sensing or contact-based constraints, increasing implementation complexity. For this reason, the central research question addressed in this work was whether controller-generated quantities could provide sufficiently informative and stable constraints to support or partially replace classical measurement sources in the state estimator to bound inertial drift.

The work was structured progressively. First, the real IMU sensor was characterized experimentally in order to assess noise levels, bias behavior and practical limitations. Initial IMU-only experiments were conducted, including testing of Zero Velocity Update (ZUPT) strategies. These preliminary studies allowed a deeper understanding of inertial drift mechanisms and of the sensitivity of the estimator to bias and noise.

The focus was then shifted to quadruped state estimation in simulation. Two alternative fusion strategies combining IMU data and NMPC output were implemented and evaluated within the MuJoCo environment.

- In the first approach, the model propagation followed the classical inertial navigation mechanization formulation (IMU-based), consistent with standard inertial navigation literature; while the controller output (specifically the predicted linear and angular velocities) was used in the correction step.
- In the second approach, a dual structure was explored, in which the model was based on the NMPC output, while IMU measurements were exploited for correction.

The simulation results highlight several important aspects.

The **first method**, demonstrated satisfactory performance just when relying primarily on IMU information. The correction step based on NMPC output revealed structural weaknesses and when significant weight was assigned to controller-derived quantities, the estimator exhibited instability or even divergence. The **second method** appeared to be slightly more accurate, but only when the estimator tuning favored IMU measurements over controller information. In practice, this implies that the controller output played a marginal role in constraining the estimate and also here, when its influence was increased, similar instability phenomena emerged.

Overall, both approaches revealed several structural limitations:

- The NMPC output does not provide sufficiently strong or independent constraints to effectively bound inertial drift. In practice, the inertial component remains dominant. In a real-world scenario, where IMU biases and disturbances are unavoidable, relying primarily on inertial integration without stronger physical constraints would not be sufficient to ensure long-term stability.
- The use of derived quantities (propagation model from NMPC differentiating a velocity, or the use of non-direct measurements), both in propagation and in correction, is intrinsically a fragile practice. State estimators benefit from direct, independent sensor measurements and using controller outputs, or quatities derived from these outputs introduces correlation and weakens the estimator structure.
- Most importantly, the initial idea of using the MPC output as an information source, either as model or correction, appears not fully valid. In both tested methods, assigning significant weight to NMPC quantities led to instability or performance degradation. Moreover, since the estimated state is fed back into the controller, increasing reliance on controller-derived measurements may amplify coupling effects and potentially deteriorate closed-loop robustness.
- An additional limitation of the comparative analysis between the two dual strategies lies in the strong influence of tuning. In practice, the observed differences between the two methods were often dominated by covariance tuning rather than by structural advantages of one formulation over the other. This highlights the limited robustness of both approaches.

Finally, the evaluation and validation process itself presents some limitations given that it was performed exclusively in simulation, with simplified sensor models. A more systematic and quantitative analysis using more quantitative metrics, longer trajectories, and varied

terrain conditions would further strengthen the conclusions.

5.1. Future Developments

From what discussed in this thesis, the natural next step is to move toward a more conventional and structurally robust state estimation architecture, while preserving the insights gained from the present study. A promising direction is the implementation of an IMU-driven EKF explicitly including gyroscope and accelerometer biases in the state vector, as bias estimation is essential to prevent long-term drift. More importantly, the correction step should rely on direct and independent sensor measurements rather than on controller-derived quantities.

As commonly adopted in quadruped literature, a robust solution consists in combining feet contact detection with leg kinematics to extract corrective constraints. When a foot is detected in stable contact with the ground, it can be assumed to have zero velocity in the world frame. Through forward kinematics based on joint encoders, the contact point velocity can be expressed as a function of base and joint velocities. Enforcing this zero-velocity condition during contact yields a measurement equation that directly constrains base linear velocity and orientation. When multiple feet are simultaneously in contact the estimator benefits from significantly richer and more independent information than what can be obtained from controller outputs alone. According to the current NMPC configuration, approximately 60% of the gait cycle involves three legs in contact, a percentage that can also be tuned for increased stability. The propagation would remain IMU-based, as in the first considered method, while corrections would be applied whenever reliable contact information is available.

An additional improvement would consist in adaptively fusing multiple correction sources according to their estimated uncertainty. These could include contact-based constraints, ZUPT updates during prolonged stationary phases (leveraging the IMU-only stationary detector developed in this work) and lower-rate SLAM-based pose information, once the SLAM module currently under development becomes available.

Finally, comprehensive experimental validation on the real robot, including realistic IMU biases and mechanical vibrations, will be essential to assess the practical robustness of the proposed architecture.

Bibliography

[1] T. A. Ali. Positioning with wide-area gnss networks: Concept and application. *Positioning*, 3(1):1–7, 2012. doi: 10.4236/pos.2012.31001.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley, 2001.

[3] 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 (RSS)*, Sydney, Australia, 2012.

[4] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, and R. Siegwart. State estimation for legged robots on unstable and slippery terrain. *The International Journal of Robotics Research*, 32(13):1503–1523, 2012. doi: 10.1177/0278364913495935.

[5] J. Borenstein, H. R. Everett, and L. Feng. Where am i? sensors and methods for mobile robot positioning. Technical report, University of Michigan, 1996.

[6] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016. doi: 10.1109/TRO.2016.2624754.

[7] M. Camurri, M. Fallon, M. Bjelonic, and M. Hutter. Probabilistic contact estimation and impact detection for state estimation of quadruped robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3313–3320, 2017.

[8] CERN. Robotics at cern, be-cem-mro. `https://cem-mro.web.cern.ch/home`, .

[9] CERN. Robotics at cern, be-cem-mro. `https://cem-mro.web.cern.ch/robotics-challenges-cern`, .

[10] E. Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer Graphics and Applications*, 25(6):38–46, 2005.

[11] Gable Technologies B.V. *Gable-IMU ONE-Series Product Documentation*, 2024. URL `https://gable-imu.nl/library/ONE_SERIES_Documentation.pdf`. Revision C.05.

[12] S. Gezici et al. Accurate positioning in ultra-wideband systems. *IEEE Wireless Communications*, 18(2):19–27, April 2011. doi: 10.1109/MWC.2011.5751292.

[13] P. D. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, Boston, MA, USA, 2nd edition, 2013. ISBN 978-1608070053.

[14] R. Hartley, J. Mangelson, J. Grizzle, and R. Vasudevan. Contact-aided invariant extended kalman filtering for legged robot state estimation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

[15] E. D. Kaplan and C. J. Hegarty. *Understanding GPS/GNSS: Principles and Applications*. Artech House, 3rd edition, 2017.

[16] F. L. Markley and J. L. Crassidis. *Fundamentals of Spacecraft Attitude Determination and Control*. Springer, New York, NY, USA, 2014. ISBN 978-1-4939-0801-1.

[17] J.-O. Nilsson, A. K. Gupta, and P. Händel. Foot-mounted inertial navigation made easy. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, 2012.

[18] J.-O. Nilsson, I. Skog, and P. Händel. A note on the limitations of zupts and the implications on sensor error modeling. *IEEE Sensors Journal*, 2012.

[19] M. Park. Error analysis and stochastic modeling of mems-based inertial sensors for land vehicle navigation applications. Master's thesis, University of Calgary, Calgary, Canada, 2004. URL `http://www.geomatics.ucalgary.ca/links/GradTheses.html`.

[20] M. Park and Y. Gao. Error and performance analysis of mems-based inertial sensors with a low-cost gps receiver. *Sensors*, 8(4), 2008. doi: 10.3390/s8042240. URL `https://www.mdpi.com/1424-8220/8/4/2240`.

[21] Y. L. Qingjiang Wang and X. Niu. Thermal calibration procedure and thermal characterisation of low-cost inertial measurement units. *The Journal of Navigation*, 69, 2016.

[22] J. B. Rawlings, D. Q. Mayne, and M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2nd edition, 2017.

[23] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey. Circumventing dynamic model-

ing: Evaluation of the error-state kalman filter applied to mobile robot localization. *IEEE Transactions on Instrumentation and Measurement*, 66(12):3370–3384, 2017. doi: 10.1109/TIM.2017.2754678.

[24] SBG Systems. Ins standard pipeline. `https://support.sbg-systems.com/sc/kb/latest/integrated-motion-navigation-sensors/inertial-navigation-system-ins`.

[25] D. Scaramuzza and F. Fraundorfer. Visual odometry [tutorial]. *IEEE Robotics & Automation Magazine*, 18(4):80–92, 2011. doi: 10.1109/MRA.2011.943233.

[26] P. Shirke, A. Potgantwar, and V. M. Wadhai. Analysis of rfid based positioning technique using received signal strength and directional antenna. *Positioning*, 7(2): 85–94, 2016. doi: 10.4236/pos.2016.72008.

[27] I. Skog, J.-O. Nilsson, and P. Händel. Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems. In *2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–6. IEEE, 2010.

[28] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter. A unified mpc framework for whole-body dynamic locomotion and manipulation. *IEEE Robotics and Automation Letters*, 6(3):4688–4695, 2021. doi: 10.1109/LRA.2021.3077857.

[29] J. Solà. Quaternion kinematics for the error-state kalman filter. Technical report, Institut de Robòtica i Informàtica Industrial (CSIC-UPC), 2017. URL `https://arxiv.org/abs/1711.02508`.

[30] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[31] D. H. Titterton and J. L. Weston. *Strapdown Inertial Navigation Technology*. The Institution of Engineering and Technology, London, UK, 2nd edition, 2004. ISBN 978-0863413582.

[32] B. Wagstaff, V. Peretroukhin, and J. Kelly. Robust data-driven zero-velocity detection for foot-mounted inertial navigation. *IEEE Sensors Journal*, 19(21):9579–9590, 2019.

[33] O. J. Woodman. An introduction to inertial navigation. Technical Report UCAM-CL-TR-696, University of Cambridge, Computer Laboratory, 2007.

[34] XSENS Technologies B.V. *XSENS MTi-1 Series Datasheet*, 2024. URL `https://www.xsens.com/products/sensor-modules/xsens-mti-1-series`.

List of Figures

List of Tables

Acknowledgements

Acknowledgments

I would like to sincerely thank my supervisor, Prof. Marco Faroni, for his guidance, support, and availability throughout this work. His advice and academic insight were fundamental in shaping the structure of this thesis.

I am grateful to my supervisor at CERN, Chris McGreavy, for the opportunity to carry out this research within the MRO group, and to Chris and Jorge Playan Garai for their technical support, insightful discussions, and continuous guidance throughout this work.

I would like to thank the entire MRO group at CERN for the stimulating environment, collaboration, and willingness to share knowledge. Being part of the team was both professionally and personally enriching.

Finally, I am grateful to my family for their encouragement and trust throughout my academic journey.