UNIVERSITY OF CRETE
DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF SCIENCES AND ENGINEERING

Robust Nonlinear State Estimation for Humanoid Robots

by

Stylianos Piperakis

PhD Dissertation

Presented

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

Heraklion, December 2019

UNIVERSITY OF CRETE

DEPARTMENT OF COMPUTER SCIENCE

Robust Nonlinear State Estimation for Humanoid Robots

PhD Dissertation Presented

by **Stylianos Piperakis**

in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy

APPROVED BY:

Author: Stylianos Piperakis

Supervisor: Panos Trahanias, Professor, University of Crete

Committee Member: Dimitris Tsakiris, Research Director, Foundation for Research and Technology - Hellas

Committee Member: Evangelos Papadopoulos, Professor, National Technical University of Athens

Committee Member: Nikolaos Tsagarakis, Senior Researcher Tenured, Italian Institute of Technology

Committee Member: Randy Gomez, Senior Scientist, Honda Research Institute Japan

Committee Member: Dimitrios Kanoulas, Lecturer, University College London

Committee Member: Maren Bennewitz, Professor, University of Bonn

Department Chairman: Angelos Bilas, Professor, University of Crete

Heraklion, December 2019

to my father...

Acknowledgments

First of all, I would like to express my sincere gratitude to my supervisor Prof. Panos Trahanias for his continuous guidance and support throughout all my years at the Computational Vision and Robotics Laboratory (CVRL). His constant encouragement and enthusiasm motivated me in all the time of research and writing of this thesis.

The two members of the advisory committee, Dr. Dimitris Tsakiris and Prof. Evangelos Papadopoulos, are also highly acknowledged for their contributions in the course of this thesis. In addition, I thank the members of the defense committee for their valuable comments and fruitful discussion in my Ph.D. presentation. In particular, I acknowledge Dr. Nikolaos Tsagarakis and Lecturer Dimitrios Kanoulas for offering access to the Instituto Italiano di Tecnologia (IIT) robotic platforms, namely WALK-MAN, COGIMON, and CENTAURO, to experimentally validate certain algorithms of my thesis.

Both the Foundation for Research and Technology-Hellas (FORTH) and the University of Crete, where this thesis was conducted, have provided a wealth of academic stimuli as well as a pleasant and creative environment for which I am heartfully thankful. Furthermore, I gratefully acknowledge the financial support provided by FORTH all these years.

I would also like to personally thank all my fellow colleagues at CVRL for the friendly atmosphere, the useful discussions, ideas and inputs that have positively influenced my work.

Last but not least, thanks to my family and my friends for the spiritual support in the highs and the lows of this work.

Abstract

Center of Mass (CoM) estimation realizes a crucial role in legged locomotion. Most walking pattern generators and real-time gait stabilizers commonly assume that the CoM position and velocity are available for feedback. In this thesis we present one of the first 3D-CoM state estimators for humanoid robot walking. The proposed estimation scheme fuses effectively joint encoder, inertial, and feet pressure measurements with an Extended Kalman Filter (EKF) to accurately estimate the 3D-CoM position, velocity, and external forces acting on the CoM. Furthermore, it directly considers the presence of uneven terrain and the body's angular momentum rate and thus effectively couples the frontal with the lateral plane dynamics, without relying on feet Force/Torque (F/T) sensing.

Nevertheless, it is common practice to transform the measurements to a world frame of reference and estimate the CoM with respect to the world frame. Consequently, the robot's base and support foot pose are mandatory and need to be co-estimated. To this end, we extend a well-established in literature floating mass estimator to account for the support foot dynamics and fuse kinematic-inertial measurements with the Error State Kalman Filter (ESKF) to appropriately handle the overparametrization of rotations. In such a way, a cascade state estimation scheme consisting of a base and a CoM estimator is formed and coined State Estimation RObot Walking (SEROW). Additionally, we employ Visual Odometry (VO) and/or LIDAR Odometry (LO) measurements to correct the kinematic drift caused by slippage during walking. Unfortunately, such measurements suffer from outliers in a dynamic environment, since frequently it is assumed that only the robot is in motion and the world around is static. Thus, we introduce the Robust Gaussian ESKF (RGESKF) to automatically detect and reject outliers without relying on any prior knowledge on measurement distributions or finely tuned thresholds. Therefore, SEROW is robustified and is suitable for dynamic human environments. In order to reinforce further research endeavors, SEROW is released to the robotic community as an open-source ROS/C++ package.

Up to date control and state estimation schemes readily assume that feet contact status is known a priori. Contact detection is an important and largely unexplored topic in contemporary humanoid robotics research. In this thesis, we elaborate on a broader question: *in which gait phase is the robot currently in?* To this end, we propose a holistic framework based on unsupervised learning from proprioceptive sensing that accurately and efficiently addresses this problem. More specifically, we robustly detect one of the three gait-phases, namely Left Single Support (LSS), Double Support (DS), and Right Single Support

(RSS) utilizing joint encoder, IMU, and F/T measurements. Initially, dimensionality reduction with Principal Components Analysis (PCA) or autoencoders is performed to extract useful features, obtain a compact representation, and reduce the noise. Next, clustering is performed on the low-dimensional latent space with Gaussian Mixture Models (GMMs) and three dense clusters corresponding to the gait-phases are obtained. Interestingly, it is demonstrated that the gait phase dynamics are low-dimensional which is another indication pointing towards locomotion being a low dimensional skill. Accordingly, given that the proposed framework utilizes measurements from sensors that are commonly available on humanoids nowadays, we offer the Gait-phase Estimation Module (GEM), an open-source ROS/Python implementation to the robotic community.

SEROW and GEM have been quantitatively and qualitatively assessed in terms of accuracy and efficiency both in simulation and under real-world conditions. Initially, a simulated robot in MATLAB and NASA's Valkyrie humanoid robot in ROS/Gazebo were employed to establish the proposed schemes with uneven/rough terrain gaits. Subsequently, the proposed schemes were integrated on a) the small size NAO humanoid robot v4.0 and b) the adult size WALK-MAN v2.0 for experimental validation. With NAO, SEROW was implemented on the robot to provide the necessary feedback for motion planning and real-time gait stabilization to achieve omni-directional locomotion even on outdoor/uneven terrains. Additionally, SEROW was used in footstep planning and also in Visual SLAM with the same robot. Regarding WALK-MAN v2.0, SEROW was executed onboard with kinematic-inertial and F/T data to provide base and CoM feedback in real-time. Furthermore, VO has also been considered to correct the kinematic drift while walking and facilitate possible footstep planning. GEM was also employed to estimate the gait phase in WALK-MAN's dynamic gaits.

Summarizing, a robust nonlinear state estimator is proposed for humanoid robot walking. Nevertheless, this scheme can be readily extended to other type of legged robots such as quadrupeds, since they share the same fundamental principles.

Keywords: Humanoid Robots, Nonlinear CoM State Estimation, Nonlinear Base State Estimation, Outlier Detection, Gait-phase Estimation, Kalman Filtering, Unsupervised Learning.

Περίληψη

Η εκτίμηση του Κέντρου Μάζας (CoM) διαδραματίζει κρίσιμο ρόλο στη ρομποτική βάδιση. Οι περισσότεροι σχεδιαστές κίνησης και ελεγκτές βάδισης πραγματικού χρόνου υποθέτουν ότι η θέση και η ταχύτητα του CoM είναι διαθέσιμες για ανατροφοδότηση ανά πάσα στιγμή. Σε αυτή τη διατριβή παρουσιάζουμε έναν από τους πρώτους τρισδιάστατους εκτιμητές κατάστασης CoM για το περπάτημα των ανθρωποειδών ρομπότ. Ο προτεινόμενος εκτιμητής συνδυάζει αποτελεσματικά τις μετρήσεις από αισθητήρες πίεσης στα πόδια, κωδικοποιητές στις αρθρώσεις και αδρανειακής μονάδας (IMU) στο σώμα με ένα Εκτεταμένο Φίλτρο Κάλμαν (EKF) για την ακριβή εκτίμηση τόσο της θέσης και της ταχύτητας του CoM αλλά και των εξωτερικών δυνάμεων που δρουν πάνω σε αυτό. Επιπλέον, λαμβάνει υπόψιν την ανωμαλότητα του εδάφους και την στροφορμή του σώματος με αποτέλεσμα να συνδυάζει το μετωπικό με το πλευρικό επίπεδο κίνησης, χωρίς να βασίζεται σε αισθητήρες δύναμης / ροπής (F/T) στα πόδια.

Ωστόσο, είναι κοινή πρακτική να επιχειρείται η μετατροπή των μετρήσεων σε ένα αδρανειακό σύστημα αναφοράς ώστε η εκτίμηση του CoM να γίνεται σε σχέση με αυτό. Κατά συνέπεια, για την επίτευξη του παραπάνω είναι υποχρεωτικό να συνεκτιμηθούν η βάση και το πόδι στήριξης του ρομπότ. Για το σκοπό αυτό, επεκτείνουμε έναν καθιερωμένο στη βιβλιογραφία εκτιμητή αιωρούμενης μάζας με τη δυναμική του ποδιού στήριξης χρησιμοποιώντας μετρήσεις κινηματικής και αδρανειακής μονάδας με το Φίλτρο Κάλμαν Σφάλματος Κατάστασης (ESKF) για την κατάλληλη διαχείριση της υπερ-παραμετροποίησης των περιστροφών. Με αυτό το τρόπο, δημιουργείται ένα σύστημα σειριακής εκτίμησης κατάστασης που αποτελείται από έναν εκτιμητή βάσης και έναν εκτιμητή CoM το οποίο ονομάζουμε State Estimation RObot Walking (SEROW). Επιπλέον, για να διορθώσουμε την κινηματική απόκλιση που προκαλείται από την ολίσθηση των ποδιών κατά το περπάτημα, χρησιμοποιούμε μετρήσεις Οπτικής Οδομετρίας (VO) και/ή Οδομετρίας LIDAR (LO). Δυστυχώς, τέτοιες μετρήσεις υποφέρουν από ακραίες τιμές σε ένα δυναμικό περιβάλλον, αφού κατά τον υπολογισμό τους χρησιμοποιείται η υπόθεση ότι μόνο το ρομπότ βρίσκεται σε κίνηση και ο κόσμος γύρω του είναι στατικός. Για αυτό το λόγο, εισάγουμε το Σθεναρό Γκαουσιανό Φίλτρο Κάλμαν Σφάλματος Κατάστασης (RGESKF) για την αυτόματη ανίχνευση και απόρριψη των ακραίων μετρήσεων. Το προτεινόμενο φίλτρο δεν βασίζεται σε πρότερη γνώση σχετικά με τις κατανομές των μετρήσεων και δεν χρησιμοποιεί ειδικά ρυθμισμένα κατώφλια. Ως εκ τούτου, το SEROW γίνεται ένα σθεναρό σύστημα

εκτίμησης κατάστασης, κατάλληλο για δυναμικά ανθρώπινα περιβάλλοντα. Προκειμένου να ενισχυθούν περαιτέρω οι ερευνητικές προσπάθειες, το SEROW δίνεται ελεύθερα στη ρομποτική κοινότητα ως ένα πακέτο ROS/C++ ανοικτού κώδικα.

Τα σύγχρονα συστήματα ελέγχου και εκτίμησης κατάστασης ανθρωποειδών ρομπότ υποθέτουν ότι η κατάσταση επαφής ποδιών-εδάφους είναι γνωστή εκ των προτέρων. Η ανίχνευση τέτοιων επαφών είναι ένα σημαντικό και σε μεγάλο βαθμό ανεξερεύνητο θέμα στη σύγχρονη ρομποτική έρευνα. Σε αυτή τη διατριβή, διατυπώνουμε μια ευρύτερη ερώτηση: σε ποια φάση βάδισης βρίσκεται το ρομπότ; Προς το σκοπό αυτό, προτείνουμε ένα ολιστικό πλαίσιο βασισμένο σε μη-επιβλεπόμενη μάθηση από δεδομένα ιδιοδεκτικής αίσθησης που αντιμετωπίζει με ακρίβεια και αποτελεσματικότητα αυτό το πρόβλημα. Συγκεκριμένα, ανιχνεύουμε με ακρίβεια μια από τις τρεις φάσεις βάδισης, την Αριστερή Υποστήριξη (LSS), την Διπλή Υποστήριξη (DS) και τη Δεξιά Υποστήριξη (RSS), χρησιμοποιώντας μετρήσεις από κωδικοποιητές, IMU και F/T. Αρχικά, πραγματοποιείται μείωση των διαστάσεων με Ανάλυση Κύριων Στοιχείων (PCA) ή με αυτόματους κωδικοποιητές ώστε να εξαχθούν χρήσιμα χαρακτηριστικά, μια συμπαγής αναπαράσταση και να μειωθεί ο θόρυβος στα δεδομένα. Στη συνέχεια, πραγματοποιείται μια ομαδοποίηση στον χώρο χαμηλών διαστάσεων με Γκαουσιανά Μοντέλα Μίγματος (GMMs). Ως αποτέλεσμα λαμβάνονται τρία πυκνά συμπλέγματα που αντιστοιχούν στις φάσεις της βάδισης. Αυτό σημαίνει ότι η δυναμική της φάσης του βαδίσματος είναι χαμηλής διάστασης το οποίο λειτουργεί ως άλλη μια ένδειξη στο ότι ολόκληρη η διαδικασία της βάδισης είναι χαμηλής διάστασης. Επιπλέον, δεδομένου ότι το προτεινόμενο πλαίσιο χρησιμοποιεί μετρήσεις από αισθητήρες που είναι συνήθως διαθέσιμοι στα σημερινά ανθρωποειδή ρομπότ, προσφέρουμε στη ρομποτική κοινότητα το Gait-Phase Estimation Module (GEM), μια ανοικτού κώδικα εφαρμογή σε ROS/Python.

Το SEROW και το GEM έχουν αξιολογηθεί ποσοτικά και ποιοτικά αφορικά με την ακρίβεια και την αποδοτικότητα τους τόσο σε προσομοίωση όσο και σε πραγματικές συνθήκες. Αρχικά, χρησιμοποιήθηκε ένα προσομοιωμένο ρομπότ στο MATLAB και το ανθρωποειδές ρομπότ Valkyrie της NASA στο ROS/Gazebo για να τεκμηριωθούν τα προτεινόμενα σχήματα στο βάδισμα πάνω σε ανομοιόμορφο/ανώμαλο έδαφος. Στη συνέχεια, τα προτεινόμενα σχήματα ενσωματώθηκαν στο α) μικρού μεγέθους ανθρωποειδές ρομπότ NAO v4.0 και β) στο πλήρους μεγέθους ανθρωποειδές WALK-MAN v2.0 για περεταίρω πειραματική επικύρωση. Με το NAO, το SEROW εφαρμόστηκε στο ρομπότ για να παράσχει την απαραίτητη ανατροφοδότηση στον σχεδιασμό της κίνησης και τη σταθεροποίηση του βηματισμού σε πραγματικό χρόνο. Με αυτό το τρόπο επιτεύχθηκε πολυκατευθυντική βάδιση ακόμη και σε εξωτερικά/ανομοιογενή εδάφη. Επιπλέον, το SEROW χρησιμοποιήθηκε στον σχεδιασμό βημάτων για την πλοήγηση και επίσης στο Visual SLAM με το ίδιο ρομπότ. Όσον αφορά το WALK-MAN v2.0, το SEROW εφαρμόστηκε με δεδομένα κινηματικής, αδρανειακής μονάδας

και F/T για να παρέχει ανατροφοδότηση βάσης και CoM σε πραγματικό χρόνο. Στην εκτίμηση λήφθηκε υπόψη και το VO για την διόρθωση της κινηματικής απόκλισης κατά το περπάτημα. Με αυτό το τρόπο διευκολύνεται σημαντικά ο πιθανός σχεδιασμός βημάτων. Τέλος, το GEM χρησιμοποιήθηκε επίσης για την εκτίμηση της φάσης της βάδισης στο δυναμικό περπάτημα του WALK-MAN.

Συνοψίζοντας, σε αυτή τη διατριβή προτείνεται ένας σθεναρός μη-γραμμικός εκτιμητής κατάστασης για το βάδισμα ανθρωποειδών ρομπότ. Παρόλα αυτά, το προτεινόμενο σύστημα μπορεί εύκολα να επεκταθεί και σε άλλους τύπους ρομπότ με πόδια, όπως τα τετράποδα, μιας και διαθέτουν τις ίδιες βασικές αρχές κίνησης.

Λέξεις κλειδιά: Ανθρωποειδή Ρομπότ, Μη-γραμμική Εκτίμηση Κατάστασης Κέντρου Μάζας, Μη-γραμμική Εκτίμηση Κατάστασης Σώματος, Ανίχνευση Ακραίων Τιμών, Εκτίμηση Φάσης Βαδίσματος, Φίλτρο Κάλμαν, Μη-επιβλεπόμενη Μάθηση.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

There is a driving force more powerful than steam, electricity and atomic energy: the Will.

Albert Einstein (1879 – 1955)

The most effective type of locomotion in uneven and/or rough terrains is legged locomotion. During the past three decades, significant advances have occurred in humanoid robot gait control and estimation. This has been motivated by the desire to develop advanced humanoid platforms and, in turn, help people in their daily lives, assist the ones with disabilities to walk, and even replace humans in hazardous environments. Accordingly, for robots to operate in structured or unstructured human environments, they need to mimic our morphology and effectively utilize their legs.

Inspired by the Fukushima disaster and the lack of disaster-response robots, the DARPA Robotic Challenge (DRC) tested humanoid robots within a range of tasks that might be needed in emergency situation, such as driving cars, opening doors, walking over rough terrain and climbing stairs. Yet the DRC showed just how far the humanoids are from truly being as useful as we'd like, or maybe even as we would imagine. For a humanoid to be successful in the challenge it must be able to constantly walk and maintain its balance over unknown types of terrain. Nevertheless, that was not the case for many humanoids participating in the challenge.

Generating robust and stable omni-directional gaits for humanoid robots can be very demanding. The major difficulties can be summarized as follows:

- **Limb coordination**. Humanoid robots are high degree of freedom mechanisms, and consequently, coordination of their links to achieve stable dynamic walking is far from trivial.

- **Hybrid nature of locomotion**. The presence of the impact in foot touchdown and liftoff leads to a hybrid by nature system, consisting of multiple continuous and dis-

crete phases. Instantaneous discrete phases arise when feet impact or liftoff the ground, whereas ordinary differential equations describe the evolution of locomotion during continuous phases.

- **Underaction**. During the single support phase, namely one foot touching the ground, humanoid robots have fewer actuators than degrees of freedom.

- **Overactuation**. During the double support phase e.g. both feet on the ground, humanoid robots have fewer degrees of freedom than actuators. Thus, the control input corresponding to a specific trajectory in the state space is not unique.

- **Unilateral multi-contacts**. Forces acting on the feet can only push the humanoid upwards and not pull on the ground. Therefore, arbitrary motions are not possible and questions such as: where to place the feet, how hard to push, or in which direction to move the body must be explicitly considered.

1.1 The Landscape of Humanoid Locomotion

Up-to-date, stable omni-directional walking constitutes a challenging and open research problem worldwide. Towards that direction numerous conditions to characterize the stability of the gait were proposed. The latter were then properly considered for walking pattern generation and real-time gait stabilization. In the current section, the most important stability criteria are briefly introduced and contemporary gait planning and control schemes are outlined.

Evidently, the Center of Mass (CoM) dynamics naturally arise in stability analysis and subsequently in motion planning and control. To this end, the accuracy, effectiveness, and robustness of the latter are directly linked to the accuracy of the employed CoM feedback.

When considering omni-directional walking over flat terrain, the CoM dynamics in the x and y axes are deemed sufficient for the task at hand. Nevertheless, when locomotion over rough and/or uneven terrain is considered, the 3D-CoM dynamics must be taken into account. Thus, quantities such as the 3D-CoM position, velocity, and external forces on the CoM are of vital importance and must be accurately estimated.

1.1.1 Stability Analysis

Since the origins of this research area, the formulation of dynamic stability criteria for obtaining a stable walking gait has always been a main point of interest. The most popular and widely-accepted criterion, the Zero Moment Point (ZMP), has been proposed by Vukobratovic et al. [1] and defined as the point inside the support polygon where the horizontal ground reaction moments vanish. In section 8.1 (Appendix A) a brief presentation for the

ZMP is given. Although ZMP has proven very successful even with the most advanced humanoid robots, other researchers have proposed new criteria for stable locomotion. The Foot Rotation Indicator (FRI) point [2] was proposed by Goswami; it extends the notion of ZMP, in the sense that it is not necessary for the ZMP to remain inside the convex hull to obtain a stable dynamic gait. FRI is briefly presented in section 8.2. To facilitate omnidirectional walking over rough or uneven terrain, a universal stability criterion based on the Contact Wrench Cone (CWC) was proposed by Hirukawa et al. [3] and presented in section 8.2.1. Capturability is the ability of a legged system to come to a complete stop, i.e. reach a captured state. The Capturability theory is presented in section 8.3 as proposed by Koolen et. al [4]. This notion is closely related to the notion of Viability [5] which was introduced to humanoid locomotion by Wieber [6]. Finally, the Gait Sensitivity Norm (GSN), shown in section 8.4, is a metric to quantify the disturbance rejection capabilities of limit cycle walkers, as proposed by Hobbelen and Wisse [7].

1.1.2 Planning and Control Approaches

In biped locomotion, the introduction of the ZMP [1] has been very influential. Most approaches use the ZMP with concentrated-mass models, such as the Linear Inverted Pendulum Model (LIPM) [8, 9], to approximate the complex dynamics and achieve on-board execution. In this direction, predictive schemes have been proposed to control the ZMP. In [10, 11] the Preview control was used with the Cart and Table model, while in [12] the Preview control with the LIPM was employed. Moreover, in [13] the Model Predictive Control (MPC) was combined with the Cart and Table model and ZMP constraints to reinforce the stability of the gait. In [14] a sparse solution to the afore-mentioned problem was proposed whereas in [15] an approximation was formulated to facilitate real-time execution.

Pratt et al. [16] and Hof [17] independently introduced the Capture Point (CP). Intuitively speaking, the CP is the point on the floor onto which the robot has to step to come to a rest. The CP was extended to the three dimensions by Takenaka et al. [18] with the introduction of the Divergent Component of Motion (DCM). The latter was utilized by Englsberger et al. [19, 20] for 3D walking pattern generation and real-time gait control on uneven terrain. Consequently, the notion of capturability was introduced in [21, 22] defining in which states the humanoid can avoid falling by taking one or more consecutive steps.

Caron et al. [23], proposed a rough terrain walking pattern generator with the Floating-base Inverted Pendulum (FIP). Starting from the COP-based Inverted Pendulum Model (IPM) with full contact stability constraints the FIP model is derived where the ZMP is allowed to leave the contact surface while the CoM translates freely in 3D. Subsequently, walking patterns are generated with nonlinear MPC and stabilized in real-time with a constrained LQR. In [24] the capturability analysis of the LIPM was generalized with the IPM to demonstrate how 3D walking over uneven terrains can be realized with the notation of

capture inputs. Interestingly, in the zero-step capturability two balance strategies naturally arose, one by regulating the COP and another by regulating the CoM height [25].

Simplified dynamical models may be efficient and computational inexpensive in practice, but they may lack important dynamic features arising from the complete models. Westervelt and Grizzle [26, 27] treated the steady walking cycle as a periodic motion. To study the motion of a hybrid, nonlinear and typically high dimensional underactuated system is far from trivial. To this end, the authors proposed the concept of Hybrid Zero Dynamics (HZD), an idea quite similar to the Partial Feedback Linearization (PFL). This procedure leads to a feedback design process in which Poincare stability analysis is directly and insightfully incorporated. Nevertheless, successful results were only realized in the planar 2D walkers with one degree of underactuation [27,28], since when generalizing to 3D walking the HZD are unstable and further stabilization is needed [29–32]. Furthermore, the double support phase was considered instantaneous. Sadati et al. [33] introduced the double support phase in the design but again for planar walking. Only recently 3D omni-direnctional was achieved through the concept of Partial Hybrid Zero Dynamics (PHZD) [34] and demonstrated with the DURUS humanoid [35,36]. One step asyptotically stable solutions were found by solving a high-dimension constrained nonlinear program with direct collocation. Both the HZD and PHZD result in more energy efficient and dynamic gaits contrasted to stiff/rigid gait generated with the simplified models.

1.2 Motivation and Aim of this Thesis

In every motion planning and gait control scheme presented above, accurate CoM feedback is assumed. More specifically, for locomotion over flat terrain, the 2D-CoM position and velocity is sufficient. On the contrary, when omni-directional walking over rough / uneven terrain is considered, the 3D-CoM position and velocity is mandatory for feedback. Nevertheless, 3D-CoM estimators at the time this research work began were not available. At the present time this work serves as one out of three 3D-CoM estimators available worldwide that applies not only for humanoid robots but for legged robots in general.

In order to achieve sufficiently accurate 3D-CoM state estimation, multiple states (besides CoM) must be accurately co-estimated. To this end, in the contents of this thesis, the term 'state' has multiple definitions which their significance will be made clear in the chapters to come. In summary, the states to be estimated are:

- Joint State: angular position and velocity of each DoF.
- Contact State: determine which leg serves as support (contact frame).
- Base State: 3D position, velocity, and orientation of torso frame.
- CoM State: 3D position and velocity of CoM.

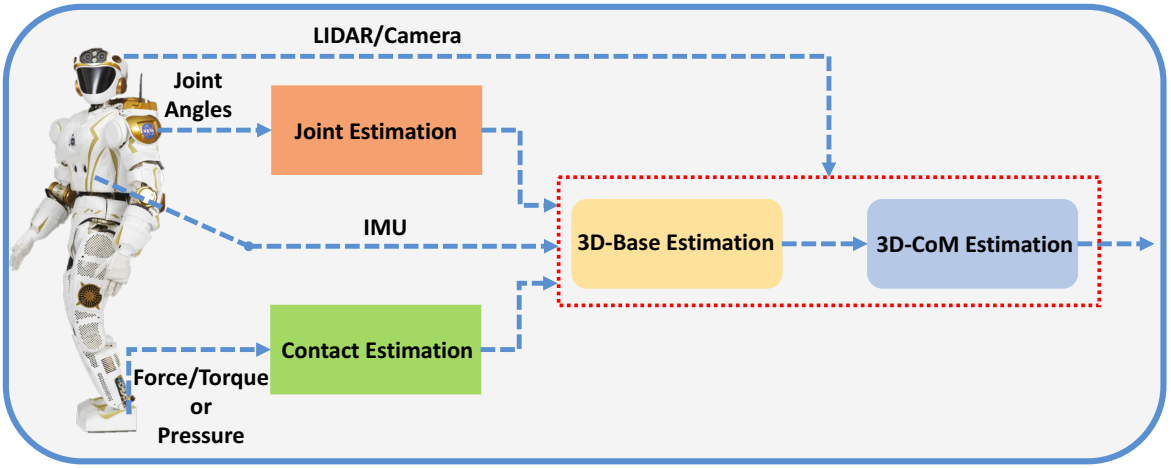

Figure 1.1: Proposed State Estimation Scheme

- External force State: 3D forces acting on the CoM.

Joint state estimation is effectively addressed by numerical differentiation followed by low-pass filtering. Therefore, will not be examined in this thesis. Nevertheless, contact, base, and eventually, CoM estimation are contemporary research topics and collectively constitute the aim of this work. The proposed state estimation scheme is conceptually illustrated in Figure 1.1. The minimum hardware requirements assumed by this work are:

- Joint encoders in every DoF.
- An Inertial Measurement Unit (IMU) on the base link.
- Pressure or Force/Torque sensors in each foot.

All sensors considered above are commonly available in contemporary humanoids and hence this work is amendable to generalization in different robotic platforms. Exteroceptive sensing, such as LIDAR or Cameras, are optionally considered to compensate for kinematic drift during the gait.

1.3 State of the Art

In this section, related work regarding humanoid, biped and quadruped robot state estimation is presented. Initially, section 1.3.1 outlines the state-of-the-art in CoM estimation. Nevertheless, as mentioned above, in order to accurately estimate the CoM, base and contact status information must be available. Thus, section 1.3.2 introduces the related work in base estimation and section 1.3.3 presents the state-of-the-art in contact detection.

1.3.1 Center of Mass Estimation

In many popular walking pattern generators and real-time gait controllers, the CoM position and velocity is needed. Towards that direction, Pongsak et al. [37] proposed one of the very first approaches. The measurements considered were provided by two linear accelerometers and a gyroscope. The estimation strategy assumed the humanoid as a single rigid body subject to an external wrench which represented the input in the rigid-body dynamics. The resulting dynamical system was linearized about the horizontal and stable position and an optimal H_2-filter for the associated estimation problem was employed.

Stephens [38] used simplified models based on the the LIPM dynamics [39] for state-estimation in order to control the posture of the force-controlled Sarcos Primus humanoid. He was able to estimate modeling errors as incoming external forces, and possible CoM biases by fusing CoM and COP measurements from the joint encoders and the Foot Sensitive Resistors (FSRs) respectively. Nevertheless, he observed that there was a trade-off between disturbance estimation and state estimation, since time-varying disturbances demanded a carefully tuning of the noise covariances. Based on that approach, Xinjilefu and Atkeson [40] compared two Kalman Filter (KF) schemes; one based on the LIPM dynamics and one based on robot's planar dynamics. They observed that LIPM KF was simple to design and implement, easy to tune, robust to modeling errors, and can generalize to other robots, while, as expected, the Planar KF yielded more accurate estimates since it is based on a more accurate representation of the robot's dynamics. Another approach based on the LIPM dynamics was presented by Kwon and Oh [41], where the current COP measurement was the input of a KF and the output was the CoM position. The filter's state was augmented with a CoM bias and a state for external forces. A similar approach but without the CoM bias in the state vector was proposed by Wittmann et al. [42], where a state estimator for biped robots fusing encoders, IMU, and force/torque measurements with a KF based on the LIPM dynamics was presented. Bae and Oh [43] considered the robot's flexibility with a compliant inverted pendulum model to enhance estimation accuracy. Experimental results were demonstrated with a simulated DRC-HUBO robot [44].

However, when the LIPM dynamics are employed, one postulates that the dynamics in the x and y axes are independent and furthermore, that the CoM lies on a constant height plane, thus only 2D-CoM estimation can be achieved. Presumably this is not the

case in real conditions, and definitely not when the robot walks on uneven ground. Hence, Carpintieri et al. [45] used a complimentary filter for 3D-CoM estimation based on consistent dynamics. The latter approach could accurately estimate the 3D-CoM position but not its velocity. Rotela et al. [46] proposed a momentum estimator for 3D-CoM position, velocity and external wrench estimation based on an Extended Kalman Filter (EKF). Nevertheless, both [45], [46] explicitly assumed that 6D-Force/Torque (F/T) sensors are employed on the robot's feet. At the same time, the 3D-CoM estimation scheme [47–49] was proposed which does not rely on feet F/T sensing and additionally considers IMU measurements in the estimation process.

1.3.2 Floating Base Estimation

Base estimation has a vital role in humanoid robot locomotion. In [50] the base orientation of the planar humanoid Rabbit was estimated. Although, orientation estimation can be effectively achieved with IMUs [51, 52], the authors in [50] conducted an observability analysis based on the existence of a state-space transformation of the nonlinear humanoid's dynamics. This transformation led to almost linear dynamics which allowed the application of state-of-the-art linear observers. Among the proposed observers similar accuracy was recorded, namely 2 degs error on the base orientation and less than 37 deg/s on the angular velocities.

Recently, Kuindersma et al. [53] proposed a base estimator based on Newton-Euler dynamics of a floating mass for estimating the body position, orientation, and velocity utilizing an IMU, the robot kinematics, and a LIDAR sensor with a Gaussian particle filter, yielding very low drift [54]. The orientation was considered as a rotation matrix linking the base to the world frame, whereas the orientation uncertainty was expressed as a screw in exponential coordinates around the base frame. This scheme was extended in [55] by considering the visually obtained terrain landscape, rendering an ATLAS robot able to walk continuously up and down staircases.

A similar approach was proposed by Bloesch et al. [56] for quadruped robots, where the IMU and the kinematic measurements were used to estimate the base motion, the contact positions, and the IMU biases. The underlying state-space included point contact dynamics, modeled as random walks, to serve as kinematic constraints. The proposed filter considered a quaternion-based representation for the base attitude to obtain a global singularity-free representation. Subsequently, the latter scheme was appropriately adapted in [57] for humanoids while also considering the feet orientation again as quaternions. State estimation was accomplished in both cases with an EKF. In addition, an interesting observability analysis was conducted by the means of the local nonlinear observability matrix. Estimation errors were 0.05 m/s on linear velocities and 0.05 rad on orientation in a 120s walking experiment.

In [58] a base estimation scheme for the HRP-2 humanoid was proposed. The authors assumed perfect kinematics for the whole body but uncertain kinematics at the ankles. The latter were modeled as series of elasticity concentrated at the robot's feet. Contact constraints where represented as virtual measurements as in [40]. The estimated state was composed of the base's angular and linear position, velocity, and acceleration and estimation was carried out with an EKF and IMU measurements. Results demonstrated a reduction on position error from 20 to 2 cm.

Xinjilefu et al. [59] examined the problem of concurrent base, joint position and velocity estimation. As before, the orientation was considered as a quaternion to avoid singularities. A Quadratic Program (QP) was formulated with joint velocities, joint torques, contact forces, and base angular velocity and linear acceleration measurements. This approach was advantageous, in the sense that it did not require a state-space model as in the KF case, could naturally handle equalities and inequalities as constraints and consider modeling error in the state vector. However, due to the imposed constraints and the high-dimensionality the framework was computationally expensive for real-time execution, did not generalize since it was based on the robot's dynamics and required joint torque sensing. Nevertheless, validation experiments were conducted on the Atlas humanoid proved that the QP estimates demonstrated good tracking performance and less delay which subsequently reduced controller chattering.

Every base estimation scheme above is subject to outlier measurements that can commonly occur in dynamic environments. To this end, in [60] we proposed the Robust Gaussian Error State Kalman Filter, that automatically detects and rejects outliers in base state estimation without relying on any prior knowledge on measurement distributions or finely tuned thresholds.

1.3.3 Contact Detection

Contemporary research approaches in motion planning, control and estimation for legged robots readily assume that contact states are known a priori. Whole body control [61–63] and gait planning [36, 64, 65] are explicitely based on contact models. Even when simplified dynamical models are employed in the design [10,66,67], the contact state is implicitly considered in the computation of the COP. Nevertheless, detecting ground contact and furthermore, determining which is the support leg, e.g. experiences a rigid contact with the ground, is non-trivial in legged robotics [68].

Base state estimators [53, 56, 57] and CoM estimators [46, 47, 49] utilize explicitly or implicitly leg odometry as a measurement in the filter's update step. Still, computing leg odometry requires information about the kinematics of the robot and the current contact state.

In [69] an active probabilistic contact sensing method was proposed, estimating also

the in contact link's shape and environment's attributes by executing compliant motions with a robotic arm. Contact estimation approaches can be classified into two categories, namely approaches that rely on kinematics and dynamics to estimate the Ground Reaction Forces (GRFs) and estimators that directly incorporate the measured GRFs.

Ortenzi et al. [70] demonstrated an approach to estimate the contact constraints enforced by the environment on the robot's motion using only joint encoder and kinematic information. Hwangbo et al. [71] proposed a probabilistic framework based on a Hidden Markov Model that effectively utilizes kinematics, differential kinematics, and dynamics to infer the contact status. This one dimensional method does not rely on force sensing but assumes that joint position, velocity, and torque information are available. Neunert et al. [72] used simple thresholding on the estimated GRFs to infer the contact state of a quadruped. Camurri et al. [73] demonstrated a logistic regression method to estimate contact probability. The estimated probabilities, along with kinematic information, were used in obtaining the base velocity measurement and variance to be fused in a kinematic-inertial state estimator. The proposed classifier was one dimensional and encoded the GRF threshold at which contact transition occurred. However, for the classifier to learn this threshold, the ground truth base velocity is needed. In addition, since no contact sensing is assumed the authors utilized joint encoder and torque measurements to dynamically compute an estimate of the expected GRF.

On the contrary, Bloesch et al. [74] utilized binary contact sensing to determine leg contact and used the contact constraints along with an outlier rejection methodology in the update step of an unscented Kalman filter to increase the filter's robustness. In [53,54] a Schmitt trigger is used to detect contact for a humanoid robot. More specifically, when the measured, by F/T sensors, vertical GRF crosses a high threshold, contact is detected and after the low threshold is crossed, contact is lost. The same approach was adopted in [49] with pressure sensors. Recently, Rotella et al. [75] proposed an unsupervised learning framework with a fuzzy C-means for estimating independently the contact probability for each one of the six DoFs for each leg. Additionally, similarly to [73], the authors presented a heuristic approach for updating the kinematic measurement covariance for base state estimation according to the estimated contact probability. The overall scheme was shown to perform well on low-friction rough terrain where slip/foot rotation can occur, since all six DoFs of the end-effector are considered. Nevertheless, in each foot 6D F/T and IMUs were assumed to be available.

In all state-of-the-art approaches, the objective was to determine whether a specific foot is in contact or not. Recently, in [76] we raised a broader question: *in which gait phase is the robot currently in?* To this end, we proposed a holistic framework, based on unsupervised learning from proprioceptive sensing that accurately and efficiently addresses this problem.

1.4 Thesis Contributions

The main contribution of this thesis is a complete robust nonlinear state estimator for humanoid robots. Accordingly, multiple contributions to CoM, base, and contact state estimation are made.

Initially, we introduce the first 3D-CoM estimator that considers the full nonlinear centroidal dynamics of the humanoid and does not rely on 6D F/T sensing in the feet. In addition, since it is common practice to estimate the CoM with respect to a world frame of reference, a base estimator that relies on Newton-Euler dynamics of a floating mass is extended to provide the mandatory base and support affine transformations. In such a way a cascade state estimation scheme is formed and coined State Estimation Robot Walking (SEROW). The latter is mathematically formulated and in detail presented in chapters 2–3. The proposed cascade estimation scheme is:

- Humanoid robot generic and facilitates real-time execution.
- Quantitatively and qualitatively assessed in terms of accuracy and performance. Primarily, a simulated robot in MATLAB and a Valkyrie robot in Gazebo are employed to establish SEROW with uneven/rough terrain gaits. Next, SEROW is integrated on a NAO robot and the WALK-MAN humanoid for experimental validation.
- Released as an open-source ROS/C++ package to reinforce further research endevours (`https://github.com/mrsp/serow`).

Subsequently, the robust Gaussian Error State Kalman Filter (RGESKF), a closed-form robust to outlier measurements base estimator, is mathematically established and presented in chapter 4. The RGESKF:

- Automatically detects and rejects outlier measurements, without relying on finely tuned thresholds or prior assumptions regarding the measurement distributions. Furthermore, automatically adjust the measurement noise when non-ideal but also non-outlier measurements arrive.
- Facilitates real-time execution and although demonstrated for humanoids it is readily amendable to generalization in other robotic platforms such as mobile robots or UAVs.
- Is quantitatively and qualitatively assessed in terms of accuracy and robustness with two actual humanoids, namely the WALK-MAN humanoid and a NAO robot, under realistic world conditions.
- Is released as an open-source ROS/C++ implementation within the SEROW framework.

Next, we introduce the Gait-Phase Estimation Module (GEM) to answer, to the best of our knowledge, for the first time the following question: *in which gait phase is the robot currently in?* The proposed framework is presented in chapter 5 and:

- Relies on Unsupervised Learning with proprioceptive sensing to accurately estimate the gait-phase e.g. Left Single Support (LSS), Right Single Support (RSS), or Double Support (DS).
- Is quantitatively and qualitatively assessed in terms of accuracy and efficiency both in simulation and under real-world conditions. Initially, GEM is employed for estimation in statically stable gaits with a simulated Valkyrie robot in Gazebo. Subsequently, the latter is employed to estimate the gait phase for the WALK-MAN dynamic gaits.
- Is released as an open-source ROS/Python package to reinforce further research endevours (`https://github.com/mrsp/gem`).

Finally, the efficiency and applicability of the proposed robust nonlinear state estimator is demonstrated with various robotic modules in chapter 6, namely with:

- Visual Simultaneous Localization and Mapping (SLAM).
- Footstep planning for navigation.
- Real-time motion generation and gait stabilization for uneven terrain walking.

1.4.1 Contributed Papers

Parts of the research work within this thesis have already been published and presented in relevant scientific fora, as follows:

- **S. Piperakis**, D. Kanoulas, N. G. Tsagarakis and P. Trahanias, "Outlier-Robust State Estimation for Humanoid Robots," Proceedings of the IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS), Macau, China, November, 2019 [60].
- **S. Piperakis**, N. Tavoularis, E. Hourdakis, D. Kanoulas and P. Trahanias, "Humanoid Robot Dense RGB-D SLAM for Embedded Devices," in IEEE International Conference on Robotics and Automation (ICRA): 3rd Full-Day Workshop "Towards Real-World Deployment of Legged Robots", Montreal, Canada, May, 2019 [77].
- **S. Piperakis**, S. Timotheatos and P. Trahanias, "Unsupervised Gait Phase Estimation for Humanoid Robot Walking," Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Montreal, Canada, May 2019 [76].

- **S. Piperakis**, M. Koskinopoulou and P. Trahanias, "Nonlinear State Estimation for Humanoid Robot Walking," IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 3347–3354, October, 2018 [49].

- **S. Piperakis**, M. Koskinopoulou and P. Trahanias, "Nonlinear State Estimation for Humanoid Robot Walking," Proceedings of the IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS), Madrid, Spain, October, 2018.

- **S. Piperakis** and P. Trahanias, "Cascade Non-Linear State Estimation for Humanoid Robot Locomotion," Proceedings of the Dynamic Walking, Mariehamn, Finland, June, 2017 [48].

- **S. Piperakis** and P. Trahanias, "Non-Linear ZMP-based State Estimation for Humanoid Robot Locomotion," Proceedings of the IEEE-RAS International Conference of Humanoid Robots (HUMANOIDS), Cancun, Mexico, November, 2016 [47]. (Finalist for the best interactive paper award)

1.4.2 Contribution to Relevant Studies

Our work has additionally provided contributions to the research pursued by peers in the Computational Vision and Robotics Laboratory at FORTH:

- S. Timotheatos, **S. Piperakis**, A. Argyros, P. Trahanias, "Vision Based Horizon Detection for UAV Navigation", Proceedings of the 27th International Conference on Robotics in Alpe-Adria Danube Region (RAAD), Patras, Greece, June, 2018 [78].

- S. Timotheatos, **S. Piperakis** and P. Trahanias, "Visual Horizon Line Detection for UAV Navigation," International Journal of Mechanics and Control, June, 2019 [79].

- M. Maniadakis, E. Hourdakis, M. Sigalas, **S. Piperakis**, M. Koskinopoulou and P. Trahanias, "Time-aware multi-agent symbiosis", Frontiers in Robotics and AI **(submitted)**.

- M. Koskinopoulou, **S. Piperakis**, P. Trahanias, "Learning from Demonstration Facilitates Human-Robot Collaborative Task Execution," Proceedings of the ACM/IEEE International Conference of Human-Robot Interaction (HRI), Christchurch, New Zealand, March, 2016 [80].

1.4.3 Contributed Open-Source Software

Currently, four major open-source software modules are offered to the community as Robot Operating System (ROS) packages, namely:

- ROS/C++: State Estimation RObot Walking (SEROW)
  https://github.com/mrsp/serow
- ROS/Python: Gait-phase Estimation Module (GEM)
  https://github.com/mrsp/gem
- ROS/C++ and CUDA: Kinect Fusion with SEROW (KFusion + SEROW)
  https://github.com/tavu/kfusion_ros
- ROS/C++: NAO humanoid robot omni-directional walk engine (nao_walk)
  https://github.com/mrsp/nao_walk

1.4.4 At the Time of this Writing

The research developed in the context of this thesis has received over 45 citations and over 3000 reads by the robotics community worldwide. In addition, the developed software has been adopted in various robotic platforms and continues to receive proper attention by the community.

1.5 Thesis Organization

This thesis is organized as follows:

- Chapter 2 presents the Nonlinear ZMP estimator, a 3D-CoM estimator for flat terrain.
- Chapter 3 introduces State Estimation RObot Walking (SEROW), a cascade 3D-CoM estimation scheme. The latter extends the previous estimator by directly considering the presence of uneven terrain and the angular momentum around the CoM.
- Chapter 4 proposes a closed form solution to robustify base estimation when outlier measurements are present.
- Chapter 5 introduces Gait-Phase Estimation Module (GEM), a hollistic approach to gait-phase estimation as an alternative solution to contact detection.
- Chapter 6 demonstrates how SEROW can be integrated in multiple robotic modules and platforms.
- Chapter 7 concludes the thesis and discusses possible future directions and emerging research topics.

- Appendix A compliments this thesis with a brief presentation of important criteria to characterize the gait stability.
- Appendix B provides all the neccessary mathematical formulations and derivations for this thesis.

Chapter 2
Nonlinear ZMP-based State Estimation

Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less.

Marie Curie (1867–1934)

This chapter presents a novel state estimation scheme for humanoid robot locomotion using an Extended Kalman Filter (EKF) for fusing encoder, Inertial Measurement Unit (IMU) and Foot Sensitive Resistor (FSR) measurements. The filter's model is based on the nonlinear Zero Moment Point (ZMP) dynamics and thus, coupling the dynamic behavior in the frontal and the lateral plane. Furthermore, it provides state estimates for variables that are commonly used by walking pattern generators and posture balance controllers, such as the Center of Mass (CoM) and the linear time-varying Divergent Component of Motion (DCM) position and velocity, in the 3D space. Dynamic modeling errors are taken into account in the acceleration level. In addition, an observability analysis for the nonlinear system dynamics and the linearized discrete-time EKF dynamics is presented. Subsequently, by utilizing ground-truth data obtained from a vicon motion capture system with a NAO humanoid robot, we demonstrate the effectiveness and robustness of the proposed scheme contrasted to the linear filters, even in the case where disturbances are introduced to the system. Finally, the proposed approach is implemented and employed for feedback to a real-time posture controller, rendering a NAO robot able to walk on an outdoors inclined pavement.

2.1 Aim and Contribution

Humanoid robot locomotion is a challenging task with many difficulties. Mainly, due to the nonlinear multi-body dynamics along with the many DoFs the humanoid robots have, the under-actuation which occurs during the gait, and the unilateral type of contact the robots experience with the ground. The nonlinear multi-body dynamics prohibit exact so-

lutions to be obtained in real-time. Therefore, many researchers approximated those dynamics with simplified models that could describe the dynamic behavior of a humanoid while walking. However, those models are based on the assumption that the robot's dynamics are decoupled in the frontal and the lateral plane, which is not true, especially when the robot exhibits highly dynamic motions. In addition, since the robot does not have a fixed base, it is under-actuated. Nevertheless, when the assumptions that a rigid type contact between the support leg and the ground along with sufficient friction exist, all the under-actuated DOFs vanish. Unfortunately, this is not the case in a realistic environment. Vertical displacement with respect to the ground can cause acceleration in the same direction which must be taken into consideration while planning or controlling the robot in order to avoid undesired ground reaction forces.

Contemporary CoM estimation schemes rely on the LIPM for the underlying estimation. Nevertheless, by using the LIPM dynamics, one assumes that the CoM is constrained to lie on a constant horizontal plane and furthermore that the motion in the x and y axes are decoupled. Unfortunately, neither holds in real world conditions and especially when a robot locomotes on uneven and/or rough terrain. In this chapter, we propose a state estimator that fuses effectively three different kind of sensors, generalizes with little to no effort to other humanoids, can be easily tuned, and yields accurate 3D estimates for important quantities in humanoid planning and control, even in the z-axis contrasted to the LIPM approaches. More specifically, we propose a novel estimation scheme with an EKF which has its dynamics based on the nonlinear ZMP [1] formulation, thus, effectively coupling the dynamic behavior in the frontal and lateral plane and fusing information from sensors that are widely available on humanoids, namely, encoders, IMU, and FSRs. This filter provides accurate estimates for variables that are commonly used by walking pattern generators and posture stabilization controllers, such as the 3D-CoM and the linear time-varying DCM [81] position and velocity, in the x, y, and z axes, as experimentally validated with a NAO humanoid robot under real world conditions.

2.2 Center of Mass Estimation

In this subsection, we present the EKF's process and measurement model which are used for the state estimation task. The dynamics are based on the nonlinear ZMP equation, where we treat the ZMP location on the plane and the vertical GRF as the input to the system and the output are the position and the acceleration of the CoM in the 3D space.

The ZMP is defined as the point on the ground at which the moments generated by the reaction forces vanish. By also considering external forces acting on the robot's body, the equations of motion are formulated as:

$$\ddot{c}_x = \frac{c_x - z_x}{c_z}(\ddot{c}_z + g) + \frac{1}{m} f_x \tag{2.1}$$

$$\ddot{c}_y = \frac{c_y - z_y}{c_z}(\ddot{c}_z + g) + \frac{1}{m} f_y \tag{2.2}$$

where z_x, z_y, f_x, f_y are the ZMP coordinates and modeling errors in the x and y axes respectively, c_x, c_y, c_z is the position of the CoM with respect to an inertial frame of reference, $\ddot{c}_x$, $\ddot{c}_y$, $\ddot{c}_z$ is the corresponding acceleration, g is the gravitational acceleration and m is the robot's mass. Furthermore, for the z-axis the dynamics are:

$$\ddot{c}_z = \frac{1}{m} f_N - g + \frac{1}{m} f_z \tag{2.3}$$

where f_N is the vertical GRF and f_z the modeling error in the z direction.
Replacing (2.3) in (2.1), (2.2), yields the following 3D nonlinear dynamics:

$$\ddot{c}_x = \frac{c_x - z_x}{m c_z}(f_N + f_z) + \frac{1}{m} f_x \tag{2.4}$$

$$\ddot{c}_y = \frac{c_y - z_y}{m c_z}(f_N + f_z) + \frac{1}{m} f_y \tag{2.5}$$

$$\ddot{c}_z = \frac{1}{m}(f_N + f_z) - g \tag{2.6}$$

2.2.1 Process Model

Assume the following state vector for the process dynamics:

$$\boldsymbol{x}_t = \begin{bmatrix} c_x & c_y & c_z & \dot{c}_x & \dot{c}_y & \dot{c}_z & f_x & f_y & f_z \end{bmatrix}^\top$$

with $\dot{c}_x$, $\dot{c}_y$, $\dot{c}_z$ the CoM velocity. Furthermore, assume the input $\boldsymbol{u}$ to the filter is the ZMP in the x and y axes along with the vertical GRF as measured by the FSRs:

$$\boldsymbol{u}_t = \begin{bmatrix} z_x^{\mathrm{FSR}} & z_y^{\mathrm{FSR}} & f_N^{\mathrm{FSR}} \end{bmatrix}^\top \tag{2.7}$$

Consequently, the process model takes the standard nonlinear form:

$$\dot{\boldsymbol{x}}_t = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\epsilon}_t) \tag{2.8}$$

where

$$\frac{d}{dt}\begin{bmatrix} c_x \\ c_y \\ c_z \\ \dot{c}_x \\ \dot{c}_y \\ \dot{c}_z \\ f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} \dot{c}_x \\ \dot{c}_y \\ \dot{c}_z \\ \frac{c_x-u_1}{mc_z}(u_3+f_z)+\frac{1}{m}f_x \\ \frac{c_y-u_2}{mc_z}(u_3+f_z)+\frac{1}{m}f_y \\ \frac{1}{m}(u_3+f_z)-g \\ 0 \\ 0 \\ 0 \end{bmatrix} + \boldsymbol{\epsilon}_t \tag{2.9}$$

and $\boldsymbol{\epsilon}_t$ is a Gaussian zero-mean additive noise with covariance $\boldsymbol{Q}_t$, $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_t)$.

Taking the appropriate continuous dynamic linearization, yields the following Jacobian matrix of the state vector $\boldsymbol{x}$:

$$\boldsymbol{G}_t = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \mathbf{0} & \boldsymbol{I} & \mathbf{0} \\ \boldsymbol{C}_t & \mathbf{0} & \boldsymbol{D}_t \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{2.10}$$

with

$$\boldsymbol{C}_t = \begin{bmatrix} \frac{u_3+f_z}{mc_z} & 0 & -\frac{(u_3+f_z)(c_x-u_1)}{mc_z^2} \\ 0 & \frac{u_3+f_z}{mc_z} & -\frac{(u_3+f_z)(c_y-u_2)}{mc_z^2} \\ 0 & 0 & 0 \end{bmatrix} \tag{2.11}$$

$$\boldsymbol{D}_t = \begin{bmatrix} \frac{1}{m} & 0 & \frac{c_x-u_1}{mc_z} \\ 0 & \frac{1}{m} & \frac{c_y-u_2}{mc_z} \\ 0 & 0 & \frac{1}{m} \end{bmatrix} \tag{2.12}$$

Although, a more accurate approximation could be used to compute the discretized matrix $\boldsymbol{G}_k$, Euler integration is used for simplicity:

$$\boldsymbol{G}_k = \boldsymbol{I} + \boldsymbol{G}_t \Delta t \tag{2.13}$$

where Δt is the sampling time.
To this end, the prediction step of the EKF is readily formulated as:

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_k, \mathbf{0})\Delta t + \hat{\boldsymbol{x}}_{k-1|k-1} \tag{2.14}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{G}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{G}_k^\top + \boldsymbol{Q}_k \tag{2.15}$$

with $\boldsymbol{P}$ being the estimate error covariance matrix.

2.2.2 Measurement Model

For the output dynamics we employ sensors that are commonly available on humanoid robots nowadays. We assume that the robot is equipped with encoders on every joint and thus we are able to compute the CoM position with respect to the torso local frame. Moreover, with the IMU we compute the corresponding CoM accelerations, again in the torso's local frame. Notice that all measurements, denoted as y_i, need to be transformed to the inertial frame of reference.

$$
\begin{aligned}
y_1 = c_x^{\text{ENC}},\ y_2 = c_y^{\text{ENC}},\ y_3 = c_z^{ENC},\\
y_4 = \ddot{c}_x^{\text{IMU}},\ y_5 = \ddot{c}_y^{\text{IMU}},\ y_6 = \ddot{c}_z^{IMU},
\end{aligned}
$$

Subsequently, since the CoM acceleration is not part of the state vector, the output equation is nonlinear:

$$
\boldsymbol{y}_t = \boldsymbol{h}(\boldsymbol{x}_t, \boldsymbol{u}_t) + \boldsymbol{\delta}_t \tag{2.16}
$$

with

$$
\boldsymbol{h}(\boldsymbol{x}_t, \boldsymbol{u}_t) = \begin{bmatrix} c_x \\ c_y \\ c_z \\ \frac{c_x - u_1}{m c_z}(u_3 + f_z) + \frac{1}{m} f_x \\ \frac{c_y - u_2}{m c_z}(u_3 + f_z) + \frac{1}{m} f_y \\ \frac{1}{m}(u_3 + f_z) - g \end{bmatrix} \tag{2.17}
$$

and $\boldsymbol{\delta}_t$ be the Gaussian zero-mean measurement noise with covariance $\boldsymbol{R}_t$, $\boldsymbol{\delta}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_t)$. After discretizing, the Jacobian matrix $\boldsymbol{H}_k = \frac{\partial h}{\partial x}$ can be readily computed, following the derivation of $\boldsymbol{G}_t$. Then, the EKF update step is realized as:

$$
\boldsymbol{K}_k = \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^\top + \boldsymbol{R}_k \tag{2.18}
$$

$$
\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{u}_k)) \tag{2.19}
$$

$$
\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)\boldsymbol{P}_{k|k-1} \tag{2.20}
$$

where $\boldsymbol{K}_k$ is the Kalman gain.

2.3 Observability Analysis

2.3.1 Nonlinear Observability Analysis

Nonlinear observability analysis is far from trivial, as it is in the linear time-invariant case, where the observability properties are excessively studied over the years and thus, well-understood. This is mainly due to the strong dependence the analysis has on the underlying nonlinear dynamics and the neighborhood of the current system's state and input. Therefore, we have included section 9.3 in appendix B to briefly present some important results from the nonlinear geometric control theory that are used in our analysis.

Following the notation introduced in the appendix B, the dimension of the state-space and the measurement's model is $n = 9$ and $m = 6$ respectively, therefore, by choosing the following coordinates $(h_1, \varphi_1^1, h_2, \varphi_2^1, h_3, \varphi_3^1, h_4, h_5, h_6)$, defined on the current operating point $(\boldsymbol{x}_t^*, \boldsymbol{u}_t^*)$, we obtain the following map:

$$\boldsymbol{\Phi}(\boldsymbol{x}_t^*, \boldsymbol{u}_t^*) = \begin{bmatrix} c_x^* \\ \dot{c}_x^* \\ c_y^* \\ \dot{c}_y^* \\ c_z^* \\ \dot{c}_z^* \\ \frac{c_x^* - u_1^*}{m c_z^*}(u_3^* + f_z^*) + \frac{1}{m} f_x^* \\ \frac{c_y^* - u_2^*}{m c_z^*}(u_3^* + f_z^*) + \frac{1}{m} f_y^* \\ \frac{1}{m}(u_3^* + f_z^*) - g \end{bmatrix} \tag{2.21}$$

By re-ordering the quantities to obtain a mathematically convenient form $\bar{\boldsymbol{\Phi}}$ and then taking the Jacobian with respect to $\boldsymbol{x}_t^*$, we get the local nonlinear observability matrix:

$$\boldsymbol{\mathcal{O}} = \frac{\partial \bar{\boldsymbol{\Phi}}(\boldsymbol{x}_t^*, \boldsymbol{u}_t^*)}{\partial \boldsymbol{x}_t^*} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{C}_t^* & \boldsymbol{0} & \boldsymbol{D}_t^* \end{bmatrix} \tag{2.22}$$

and since $\det \boldsymbol{\mathcal{O}} = \frac{1}{\mathrm{m}^3}$ we have that

$$\mathrm{rank} \boldsymbol{\mathcal{O}} = 9 \tag{2.23}$$

rendering the nonlinear dynamics in (2.8), (2.16) to be locally observable in all cases.

2.3.2 Linear Time-Varying Observability Analysis

Since, we are using a discrete EKF for the state estimation task, we must explore the observability of the filter which is based on the following linear time-varying dynamics:

$$\boldsymbol{x}_{k+1} = \boldsymbol{G}_k \boldsymbol{x}_k + \boldsymbol{\epsilon}_k \tag{2.24}$$

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{\delta}_k \tag{2.25}$$

We will give the local observability analysis based on the linear time-varying observability matrix $\boldsymbol{\mathcal{M}}$, as proposed by Chen et al. [82]. Notice, in the general case, the observability properties of a discrete linear time-varying system, can differ from the observability properties of the true underlying nonlinear continuous system. This is due to the errors that arise by the linearization and/or discretization procedure.

For the state-space and output model in (2.24), (2.25), this matrix is defined as:

$$\boldsymbol{\mathcal{M}} = \begin{bmatrix} \boldsymbol{H}_k \\ \boldsymbol{H}_{k+1}\boldsymbol{G}_k \\ \vdots \\ \boldsymbol{H}_{k+8}\boldsymbol{G}_{k+7}\dots\boldsymbol{G}_k \end{bmatrix} \tag{2.26}$$

and the sufficient condition for the local observability is:

$$\text{rank}\boldsymbol{\mathcal{M}} = 9 \tag{2.27}$$

By examining the first 9×9 submatrix of $\boldsymbol{\mathcal{M}}$:

$$\boldsymbol{\mathcal{M}}^* = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{C}_k & \boldsymbol{0} & \boldsymbol{D}_k \\ \boldsymbol{0} & \Delta t\boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \tag{2.28}$$

where $\boldsymbol{C}_k$ and $\boldsymbol{D}_k$ are the matrices in (2.11), (2.12) evaluated at the k-th discrete time instant, it is straightforward to derive:

$$\det \boldsymbol{\mathcal{M}}^* = \left(\frac{\Delta t}{m}\right)^3 \tag{2.29}$$

Therefore, the linear time-varying observability matrix $\boldsymbol{\mathcal{M}}$ is full rank and cannot drop rank under any circumstances.

2.4 Results

In the current section we outline representative results that demonstrate the effectiveness and robustness of the proposed scheme, contrasted to others well-established approaches. All conducted experiments were performed with a real NAO humanoid robot. First, the EKF is compared to a KF based on the LIPM dynamics, on ground-truth data sets obtained with a vicon motion capture system. Next, the proposed approach is employed to feedback a posture stabilization controller based on the DCM, enabling a NAO robot to walk outdoors on an inclined pavement. The filter's response is contrasted to an EKF based on Newton-Euler dynamics. In all our experiments the covariance matrices were determined experimentally and set to:

$$\boldsymbol{Q} = \mathrm{diag}(1\mathrm{e}-6\mathrm{I}, 1\mathrm{e}-5\mathrm{I}, \mathrm{I}) \tag{2.30}$$

$$\boldsymbol{R} = \mathrm{diag}(5\mathrm{e}-6\mathrm{I}, 5\mathrm{e}-4\mathrm{I}) \tag{2.31}$$

2.4.1 Evaluation on Motion Captured Data Sets

A first result regards an estimation accuracy study, in terms of the Root Mean Square Error (RMSE), contrasted to a KF based on the LIPM, as proposed by Wittman et al. [42]. The LIPM KF can estimate the following state vector:

$$\boldsymbol{x}_t = \begin{bmatrix} c_x & c_y & \dot{c}_x & \dot{c}_y & f_x & f_y \end{bmatrix}^\top \tag{2.32}$$

utilizing the CoM position and velocity, as measured by the encoders, and transformed to the inertial frame with the IMU. The employed covariance matrices are given by (2.30), (2.31), where this time the matrices-dimensions are obtained by neglecting the z-axis dynamics.

We've selected 10 ground-truth data sets, collected with a vicon motion capture system consisting of 15 infra-red cameras, and a NAO v3.3 humanoid robot [83]. To begin with, we calibrated the IMU, to remove the biases and cut off unwanted high frequencies with a low-pass filter. Then, we computed the CoM with respect to the local torso frame using kinematics and the COP using the FSR in the feet which was then transformed to the torso local frame with kinematics. Subsequently, all the acquired data were transformed to the inertial frame of reference using the estimated rotation matrix obtained by the IMU. Notice, since the NAO robot is not equipped with a 3-axis gyroscope, namely no gyro rate is available about the z-axis, the vicon yaw angle was used instead. This is why no drifting is observed in all our data, although the robot drifts in many cases. Nevertheless, this does not pose any limitation, since the same data are used as input and measurements signals for both approaches.

Since we cannot plot the response for every data set, we selected one where the motion

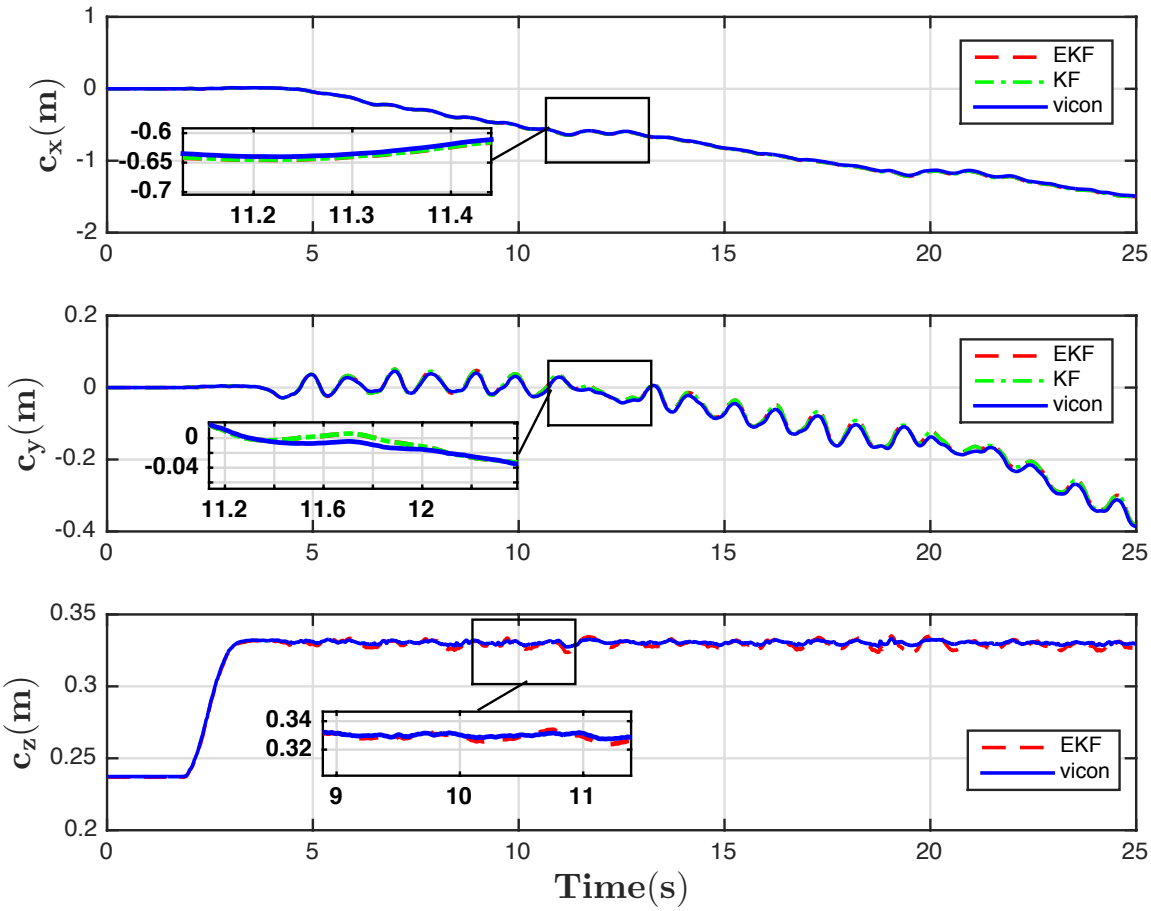

Figure 2.1: CoM position in the 3D space, blue lines indicate the ground-truth trajectories, red dotted lines the EKF estimated trajectories and green dotted the KF estimated trajectories, notice in the z axis no KF estimate is available since c_z is assumed constant.

is unstable and thus more dynamic [83]. In Figure 2.1, the CoM trajectories are shown where the EKF estimated trajectory is overlapped by the actual signal, notice in the start the robot rises from a sitting position, the EKF can accurately capture that motion. In addition, Figure 2.2 shows that the EKF estimated more accurately the corresponding CoM velocites in all axes.

Figure 2.3 illustrates the external forces/modeling error for the corresponding motion; notice that there is no delay in the force estimation as observed in the KF's case, also reported in [42]. This is due to the fusion of the acceleration measurement which yields a lag free estimation. Furthermore, notice in the z-axis, at time 0–$2s$ where the robot is practically still, the estimation is almost $15N$, this is reasonable since the NAO's FSR have a

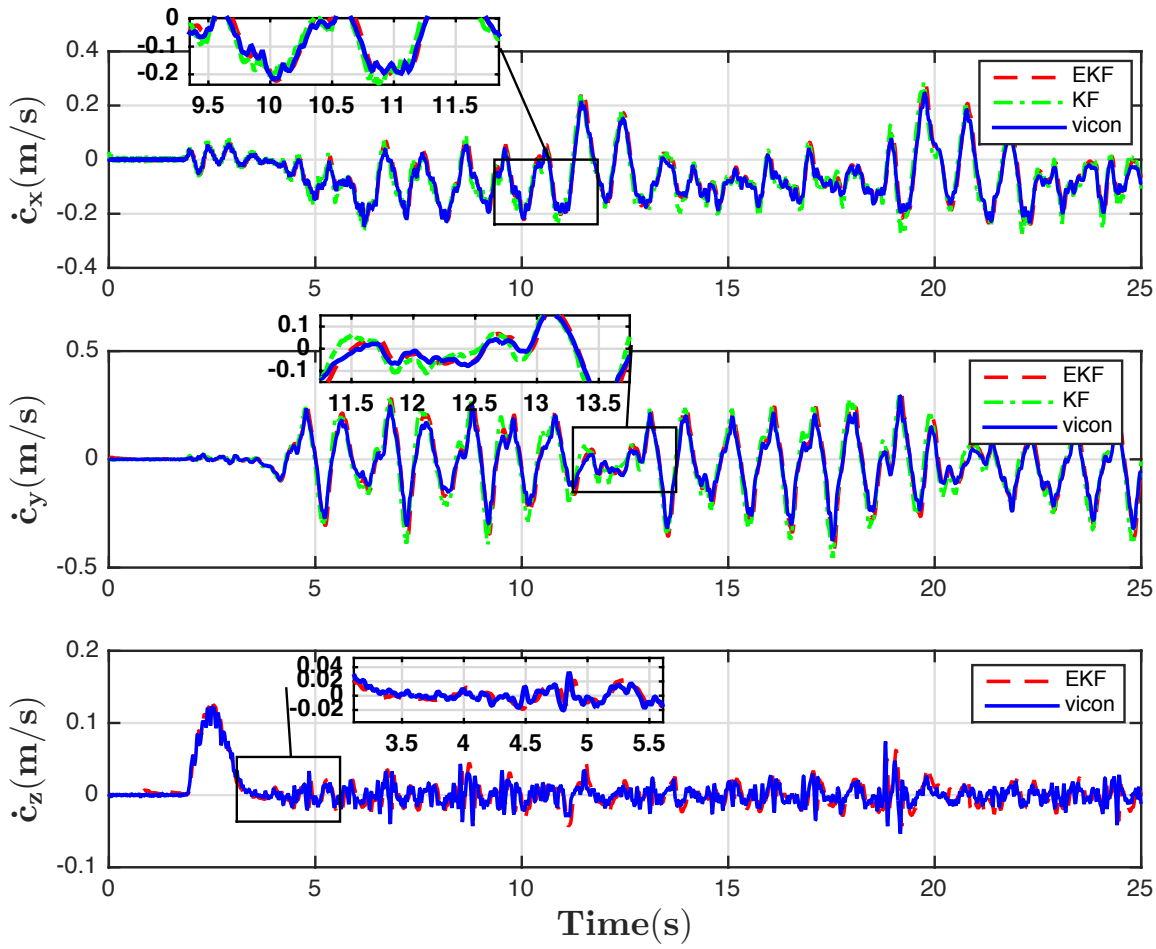

Figure 2.2: CoM velocity in the 3D space, blue lines indicate the ground-truth trajectories, red dotted lines the EKF estimated trajectories and green dotted the KF estimated trajectories, notice in the z axis no KF estimate is available since c_z is assumed constant.

reliable working range up to $25N$ and when the robot is still they measure approximately $35N$, therefore since the robot weights $4.789kg$ the modeling error needs to be approximately $1.5kg$.

Furthermore, we computed the position and velocity of the DCM. In the KF's case, we are forced to compute the Linear Time-Invariant (LTI) DCM since the assumption that the CoM lies on constant horizontal plane during the motion is made. The LTI-DCM is given by:

$$\boldsymbol{\xi}_{\text{LTI}} = \boldsymbol{c} + \frac{1}{\omega_0}\dot{\boldsymbol{c}} \tag{2.33}$$

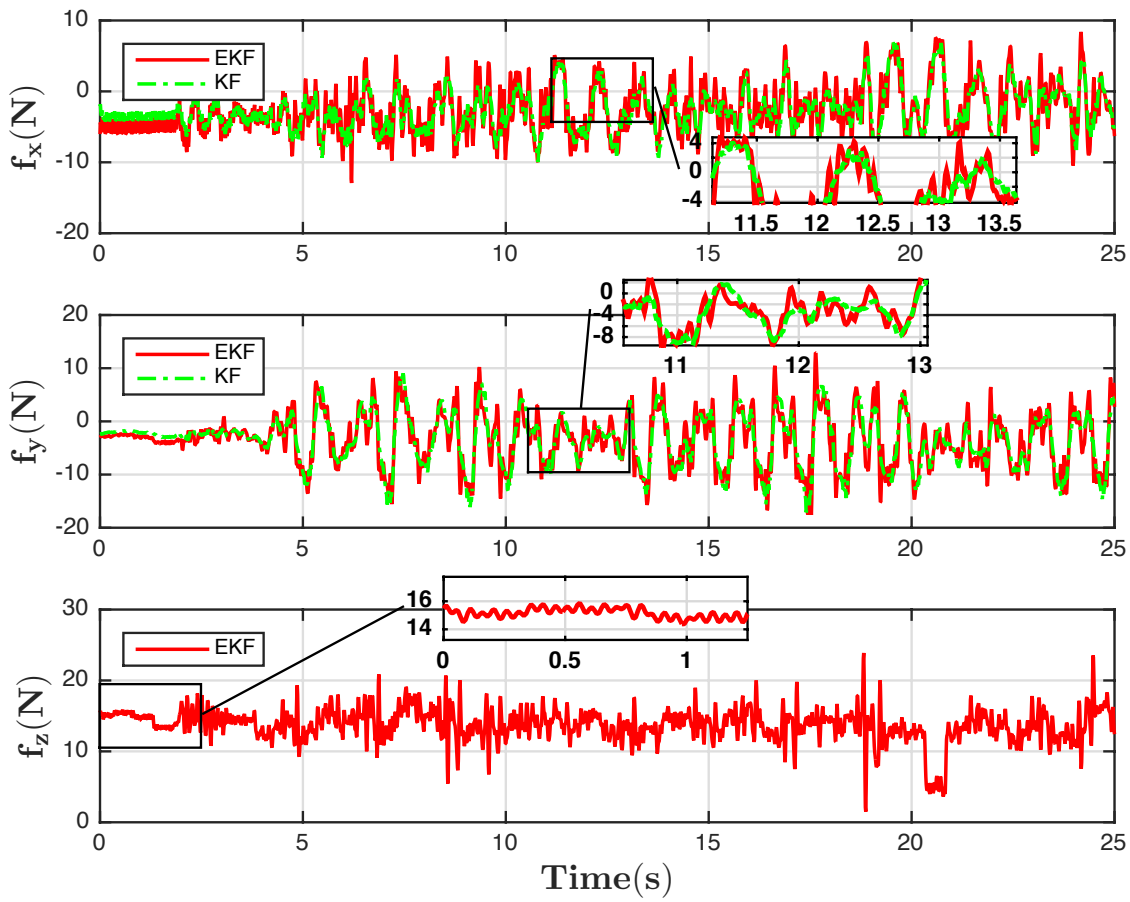

Figure 2.3: Modeling error in the x, y and z-axis respectively, KF estimates are delayed contrasted to the EKF ones.

where $\omega_0 = \sqrt{\frac{g}{l}}$ and l is the constant CoM height. The corresponding LTI-DCM velocity is thus:

$$\dot{\boldsymbol{\xi}}_{\text{LTI}} = \omega_0(\boldsymbol{\xi}_{\text{LTI}} - \boldsymbol{c}) + \frac{1}{\omega_0}\ddot{\boldsymbol{c}} \tag{2.34}$$

On the other hand, in the EKF's case we can compute the Linear Time-Varying (LTV) DCM to approximate the true nonlinear DCM more effectively. The LTV-DCM is formulated as:

$$\boldsymbol{\xi}_{\text{LTV}} = \boldsymbol{c} + \frac{1}{\omega_t}\dot{\boldsymbol{c}} \tag{2.35}$$

where $\omega_t = \sqrt{\frac{g}{c_z}}$. Since ω_t is now time-depended, the corresponding LTV-DCM velocity is given by:

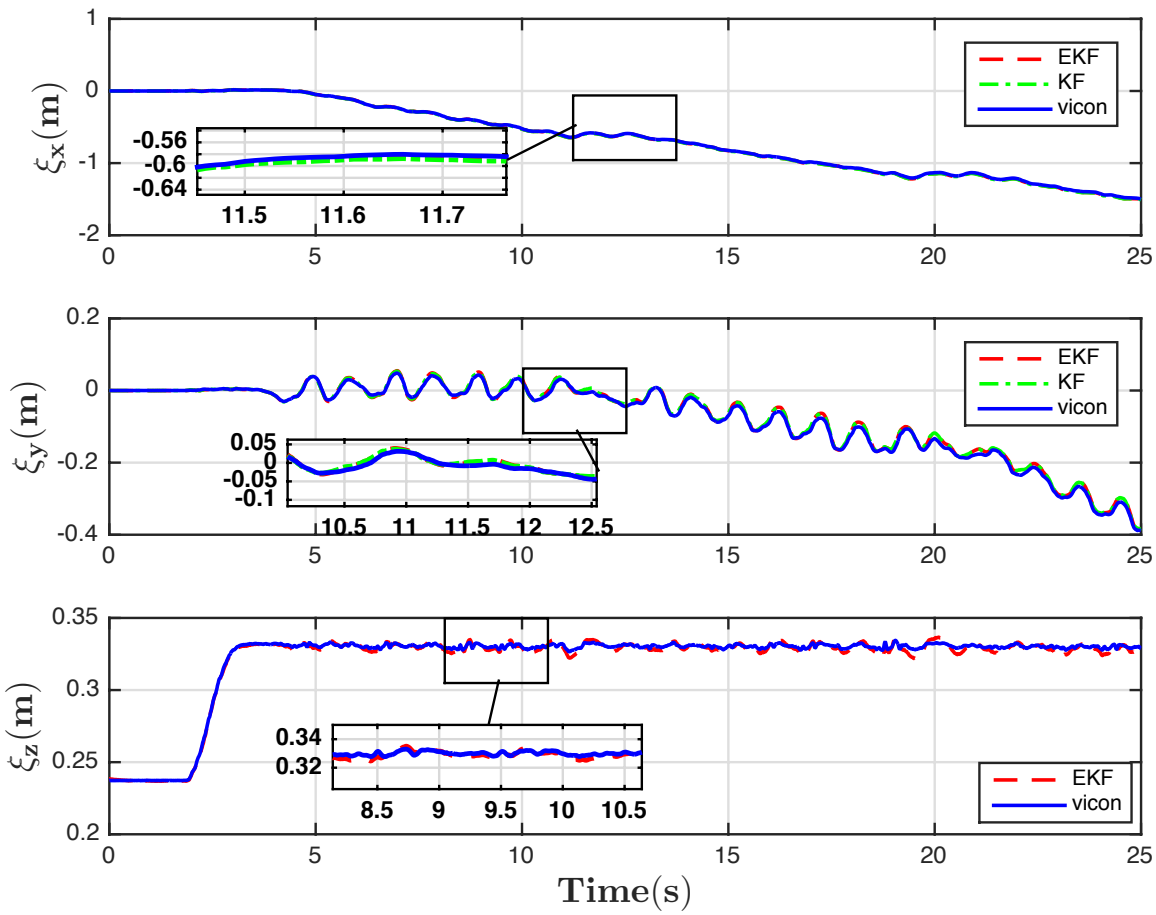

Figure 2.4: DCM position in the 3D space, blue lines indicate the ground-truth LTV-DCM, red dotted lines the EKF estimated LTV-DCM trajectories, and green dotted lines the KF estimated LTI-DCM trajectories, no KF estimate is available in the z-axis since is assumed constant.

$$\dot{\boldsymbol{\xi}}_{\text{LTV}} = \left(\omega_t - \frac{\dot{\omega}_t}{\omega_t}\right)(\boldsymbol{\xi}_{\text{LTV}} - \boldsymbol{c}) + \frac{1}{\omega_t}\ddot{\boldsymbol{c}} \tag{2.36}$$

with $\dot{\omega}_t = -\frac{g^{1/2}}{2c_z^{3/2}}\dot{c}_z$.

In both DCM velocity cases, we used the calibrated low-pass filtered acceleration by the IMU, and the corresponding CoM position and velocity estimate by each filter respectively.

Figure 2.4 shows the corresponding DCM trajectories for this case of study, while Figure 2.5 demonstrates the DCM velocities. Finally, in Figure 2.6 the average RMSE for all quantities of interest for the 10 data sets used in our study is presented. Notice that the

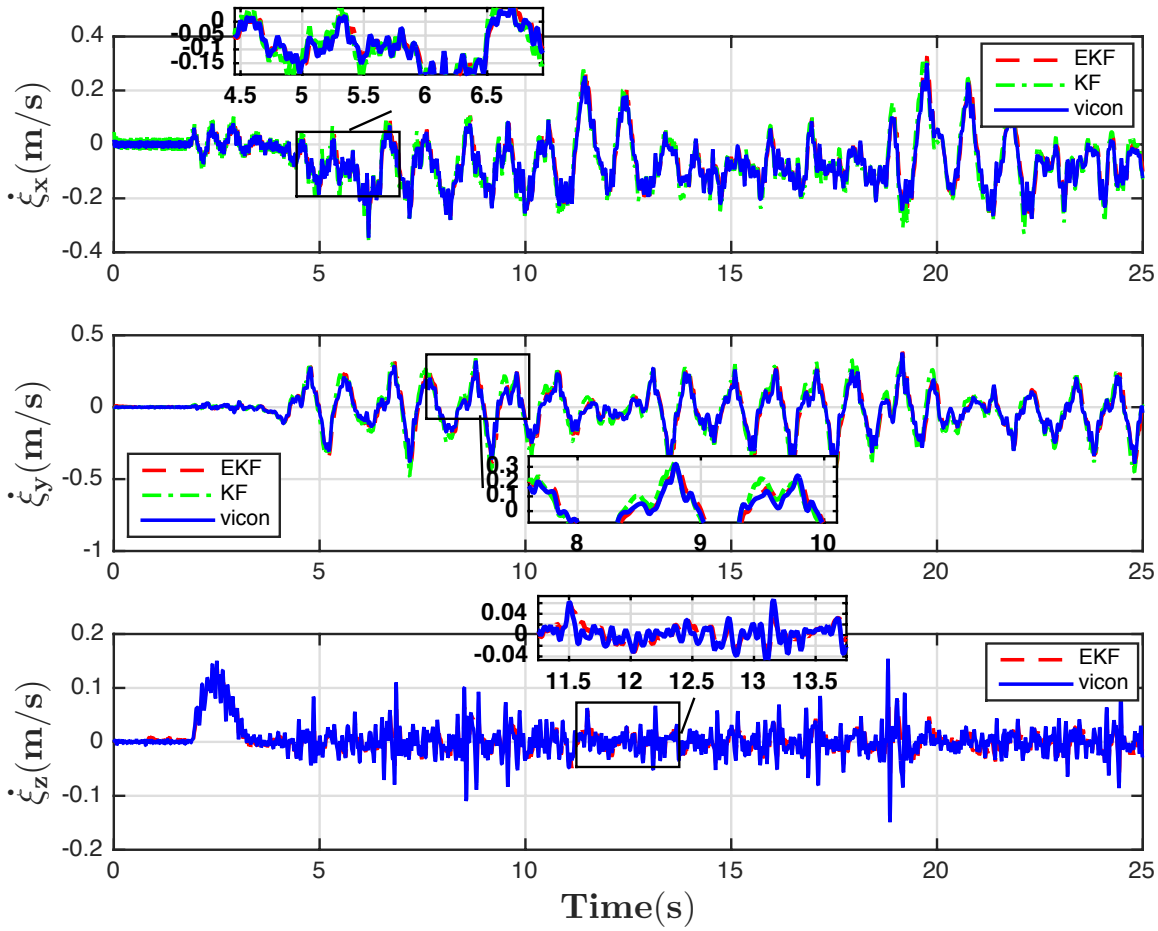

Figure 2.5: DCM velocity in the 3D space, blue lines indicate the ground-truth LTV-DCM, red dotted lines the EKF estimated LTV-DCM trajectories, and green dotted lines the KF estimated LTI-DCM trajectories, no KF estimate is available in the z-axis since is assumed constant.

EKF not only yields more accurate estimates in the RMSE sense, but also more certain ones, especially when the estimated quantities are the velocities.

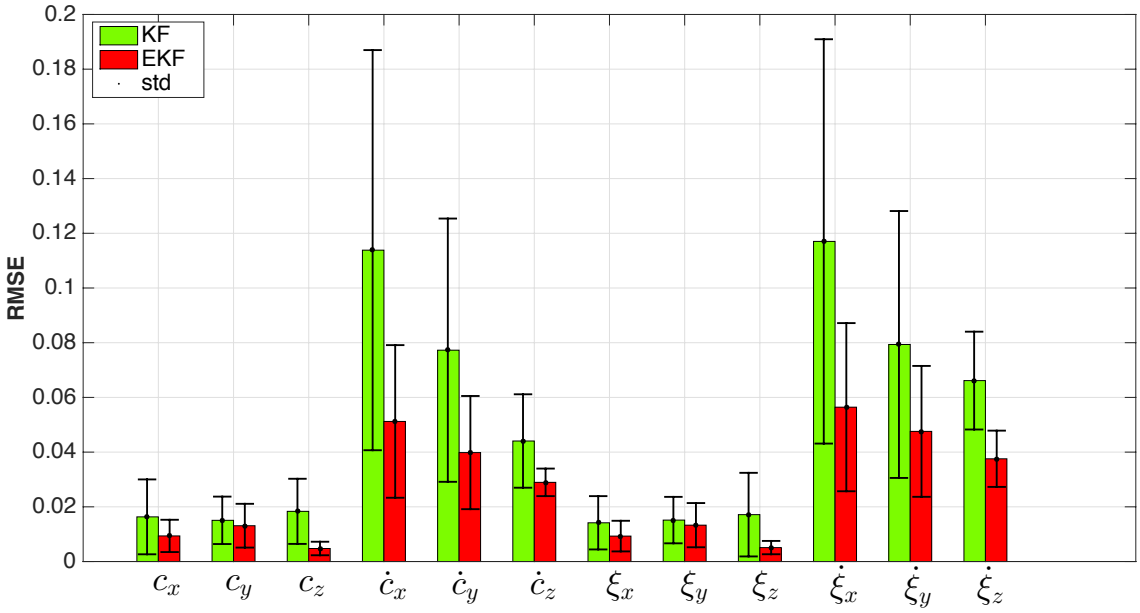

Figure 2.6: Average RMSE for all quantities of interest for the 10 data set studied; red bars indicate the EKF's accuracy, green bars the KF's accuracy, and black lines the standard deviation from the corresponding mean values.

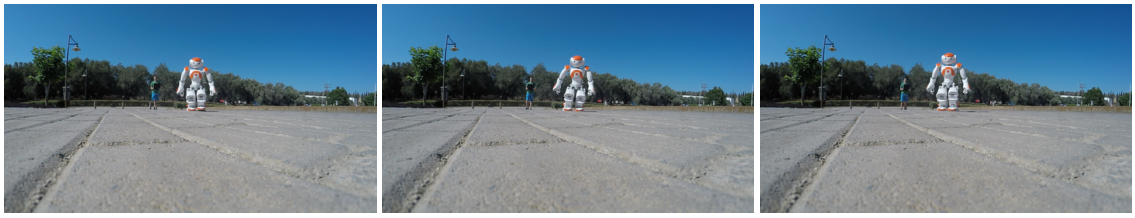

Figure 2.7: NAO humanoid robot walking diagonally on a 7° inclined pavement (from left to right).

2.4.2 Walking Outdoors on an Inclined Pavement

In this section, the proposed estimation scheme is experimentally validated on a v4.0 NAO robot. The estimates obtained by our ZMP based EKF are used to feedback a real-time posture stabilizer based on the DCM [81], rendering the robot able to keep the balance even while walking outdoors on a 7° inclined pavement, as shown in Figure 2.7. Although the slope of the ground is mild, please note that for a robot of the size of NAO it represents a rather significant challenge.

Since no ground truth data are available in outdoor environments, to verify the estimation task we implemented an EKF based on the IMU, as proposed by Rotella et.al [57], but modified in such a way to estimate the CoM quantities instead of the rigid base ones. This IMU based EKF can estimate the following state vector:

$$\boldsymbol{x}_t = \begin{bmatrix} \boldsymbol{c} & \dot{\boldsymbol{c}} & \boldsymbol{q} & \boldsymbol{b}_f & \boldsymbol{b}_\omega \end{bmatrix}^\top \tag{2.37}$$

where $\boldsymbol{q}$, $\boldsymbol{b}_f$, and $\boldsymbol{b}_\omega$, are the torso's attitude quaternion, the acceleration and the gyroscope biases respectively, utilizing the inertial CoM position as measured by the encoders. Notice, we've collected raw IMU data for 48 hours in order to perform an Alan variance analysis [84] and carefully tune the process noise covariance, as also suggested by the authors, to maximize in such a way the filter's efficiency.

Figure 2.8 and Figure 2.9 illustrate the CoM position and velocity in the 3D space as estimated by the two filters, for a diagonally forward gait on a 7° inclined pavement for approximately $25s$. Figure 2.10 shows the external force/modeling errors during the gait, notice only the ZMP based EKF can estimate those quantities. In addition, note that the magnitude of the external forces can be justified by considering that when the robot locomotes on an uneven and rough terrain, early ground contact can commonly occur, giving rise to larger external forces. Moreover, Figure 2.11 and Figure 2.12 demonstrate the LTV-DCM position and velocity respectively, as estimated by the two filters for the corresponding gait.

Furthermore, we've conducted a variety of indoors and outdoors experiments, namely,

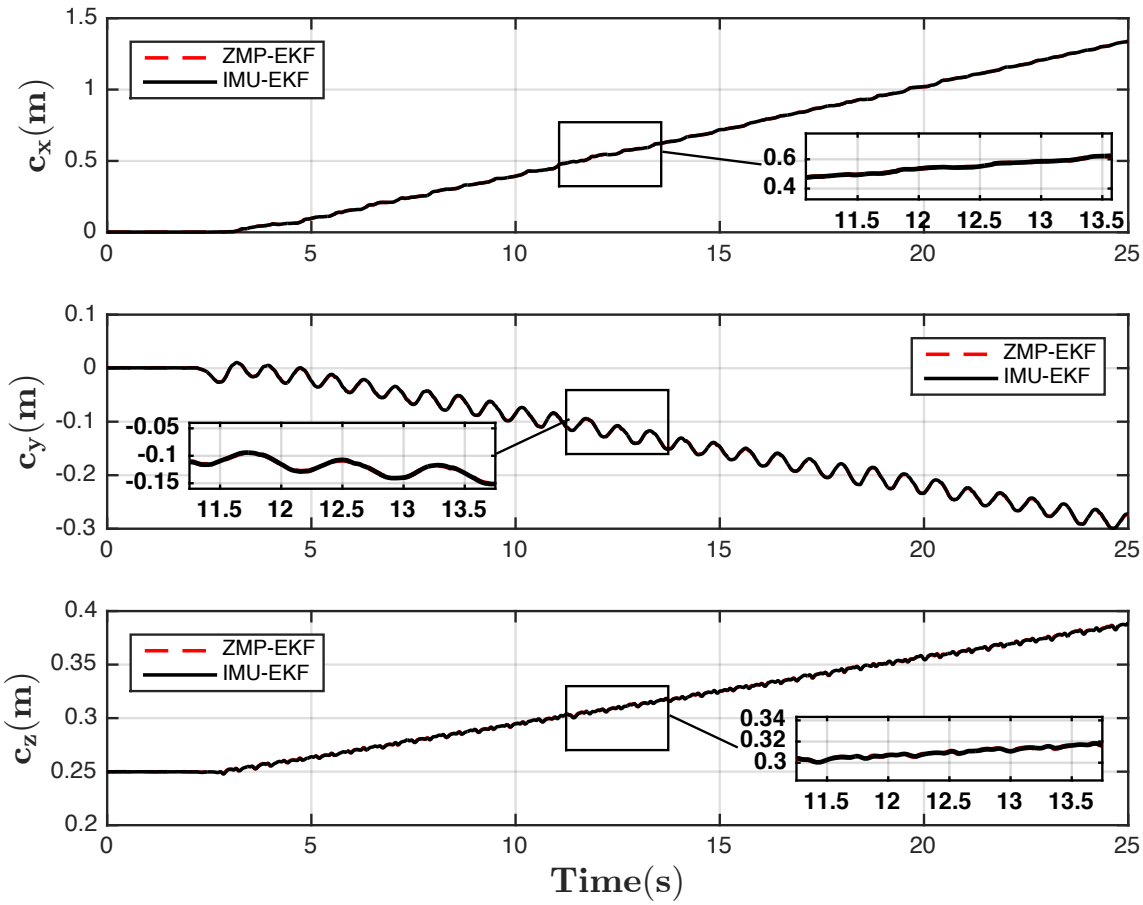

Figure 2.8: CoM position in the 3D space, red dotted lines indicate the ZMP based EKF estimated trajectories while the black lines indicate the IMU based EKF estimated trajectories.

walking indoors on a hallway, walking outdoors on an even pavement, walking in place on grass while heavily disturbing the robot and, as also illustrated in https://goo.gl/by3cB5, all the ZMP based EKF estimates were pretty similar, within noise margins, to the IMU based EKF ones, validating in such a way the proposed estimation scheme.

Notice that in all experiments reported above, the estimated z-axis components contain higher noise compared to the x and y-axis ones. This is due to the fact that the noisy FSR measurements are employed in the z-axis dynamics.

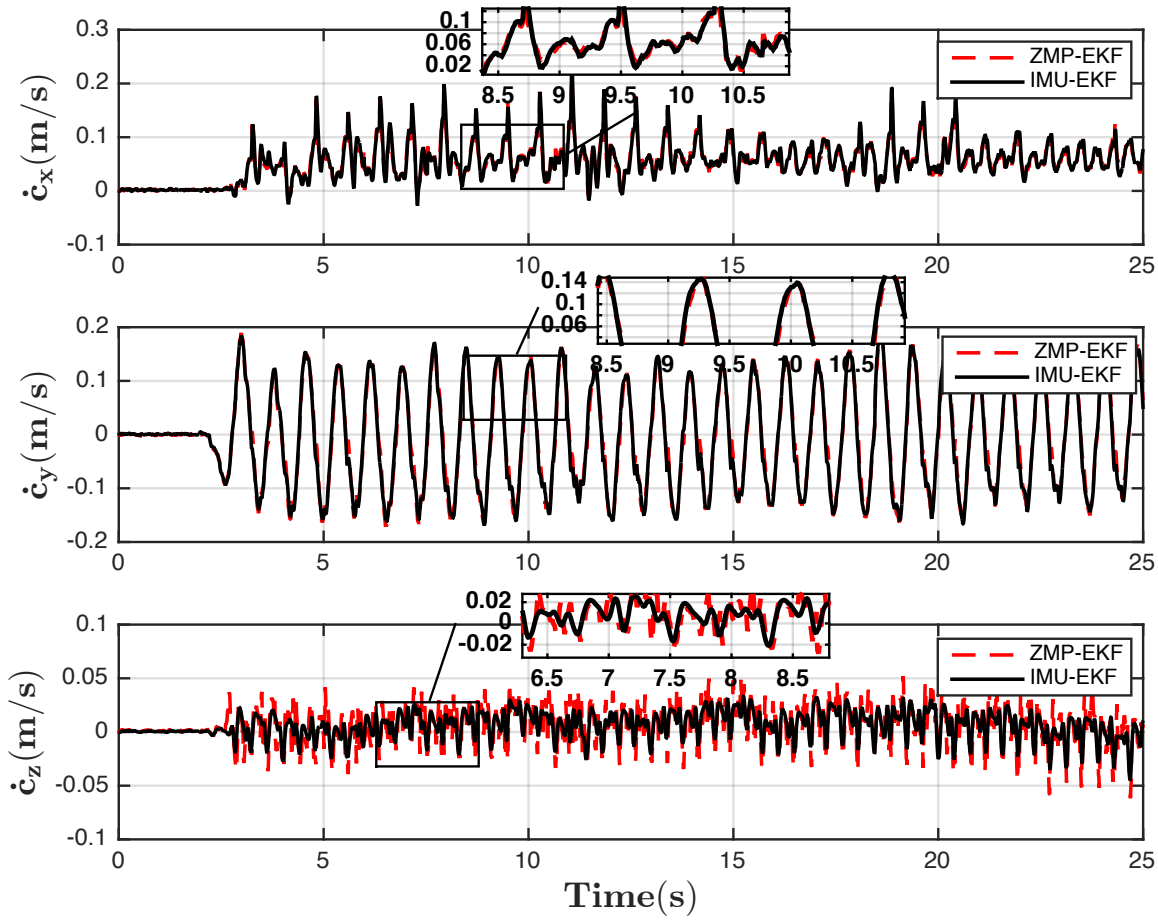

Figure 2.9: CoM velocity in the 3D space, red dotted lines indicate the ZMP based EKF estimated trajectories while the black lines indicate the IMU based EKF estimated trajectories.

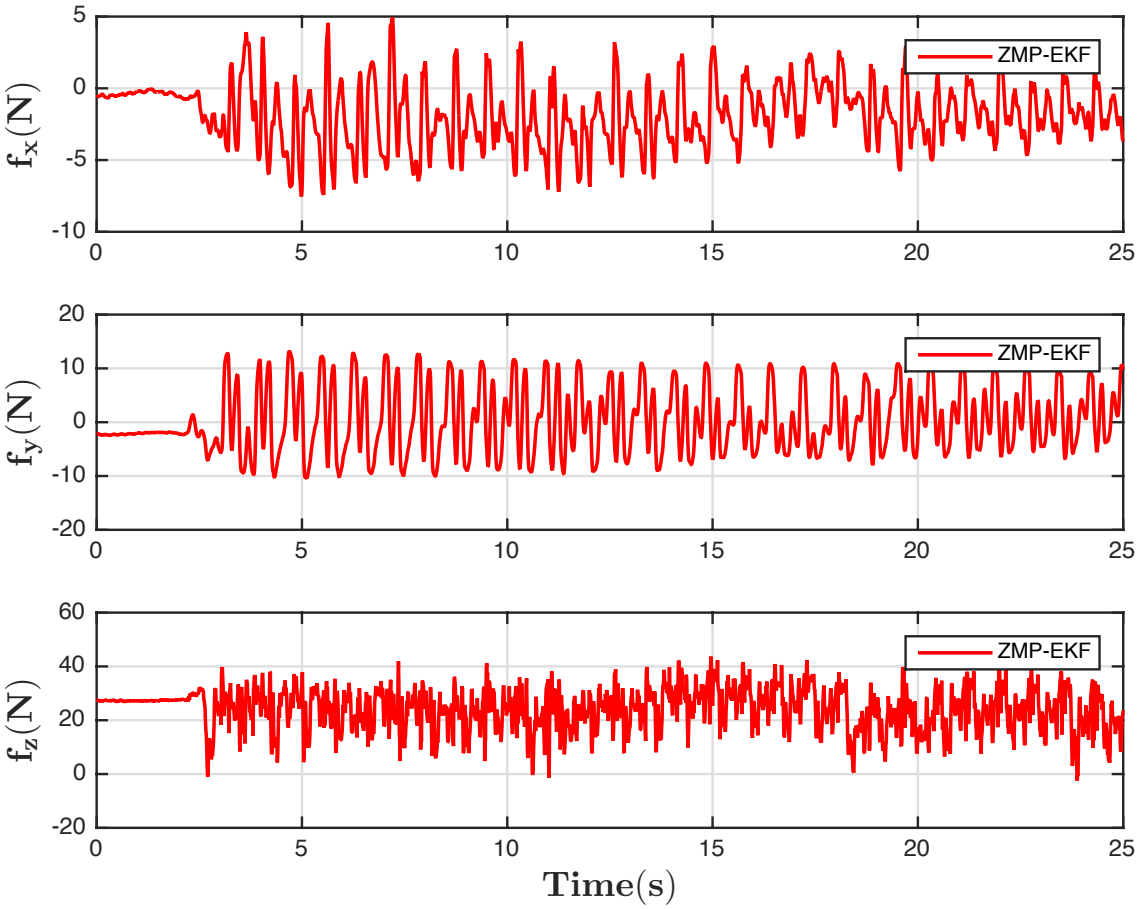

Figure 2.10: Modeling error in the x, y and z-axis respectively, IMU based EKF estimates are not available.

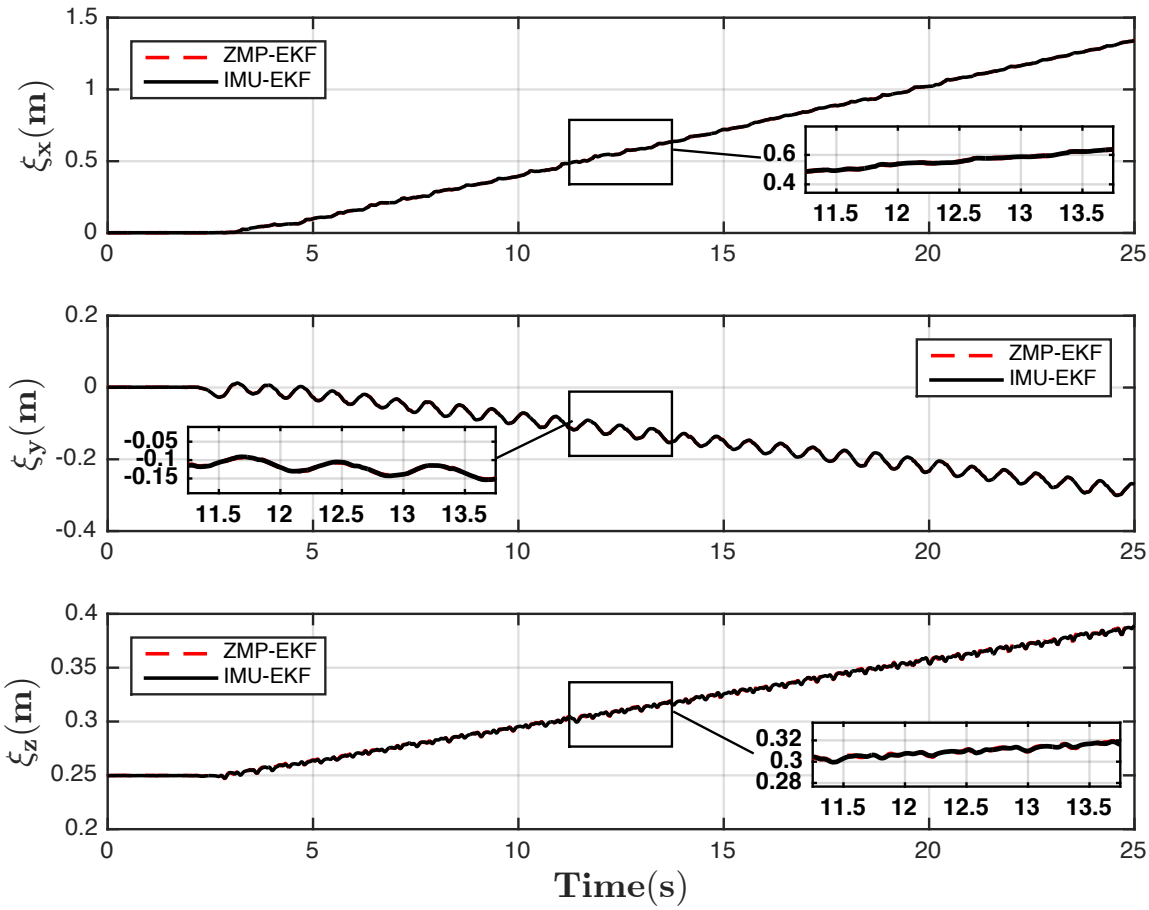

Figure 2.11: LTV-DCM position in the 3D space, red dotted lines indicate the ZMP based EKF estimates and black lines the IMU based estimates.

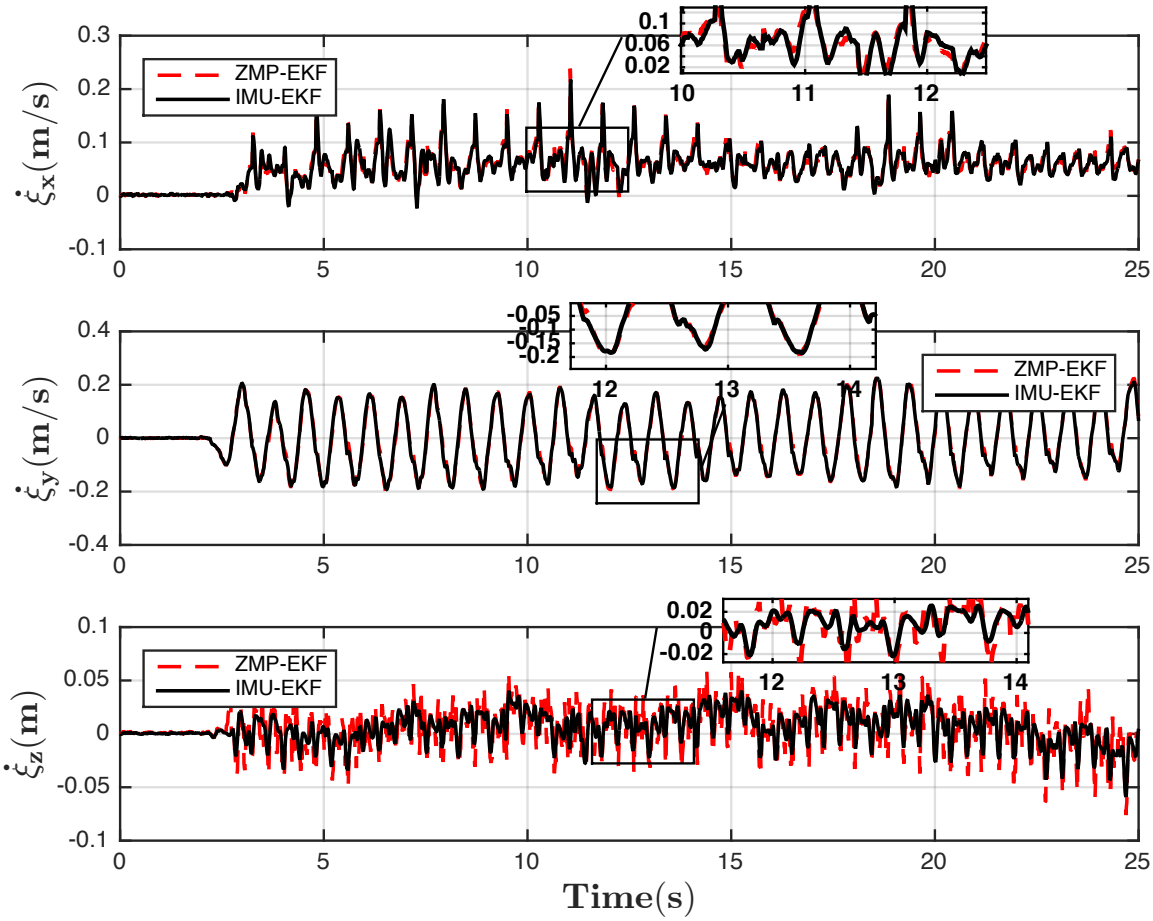

Figure 2.12: LTV-DCM velocity in the 3D space, red dotted lines indicate the ZMP based EKF estimates and black lines the IMU based estimates.

2.5 Conclusion

In this chapter, a novel state estimation scheme for humanoid robot locomotion was presented, fusing effectively three different sensor sources, namely the joint encoders, the IMU, and the FSRs. We utilized the nonlinear ZMP equation with an EKF to surpass the limitation of the constant CoM height and the planar dynamic decouple, as assumed by the LIPM, and readily estimate control variables commonly used by walking pattern generators and posture stabilization controllers. In addition, modeling errors were considered acting on the CoM in the acceleration level.

Someone would assume that the observability would be lost when the robot experience accelerations in the z-axis equal to g, e.g. the robot is in free fall. As proved by our observability analysis for both the nonlinear dynamics and the EKF this is not the case, since the local-observability matrix is full rank under all circumstances. Nevertheless, the ZMP is not well-defined when the robot is in flight.

Our experimental result demonstrated that the filter showed robustness to perturbations, quick convergence properties and provided more accurate estimates contrasted to a KF based on the LIPM. In addition, when incorporated with a real-time stabilization controller, a NAO robot was able to walk on an outdoors inclined pavement and on grass, effectively sensing and negotiating the incoming disturbances. Moreover, the filter's estimates were pretty similar to the ones obtained by an EKF based on generic rigid body dynamics and the IMU, validating the proposed approach.

In the next chapter, we will include the external forces/torques that act on the robot's CoM and the ground-height in the design, to provide more accurate estimates for the 3D-CoM position and velocity. In addition, we will outline the importance of base estimation and indicate how errors in base estimation propagate to CoM estimation.

Chapter 3
Cascade Nonlinear State Estimation

If I have seen further it is by standing on the shoulders of giants.
Isaac Newton (1642–1727)

This chapter presents a novel cascade state estimation framework for 3D-CoM estimation of walking humanoid robots. The proposed framework, called SEROW (State Estimation RObot Walking) fuses effectively joint encoder, inertial, feet pressure and visual odometry measurements. Initially, we consider the humanoid's Newton-Euler dynamics and rigorously derive the nonlinear CoM estimator. The latter accurately estimates the 3D-CoM position, velocity and external forces acting on the CoM, while directly considering the presence of uneven terrain and the body's angular momentum rate and thus effectively coupling the frontal with the lateral plane dynamics. Furthermore, we extend an established floating mass estimator to take into account the support foot pose, yielding in such a way the mandatory, for CoM estimation, affine transformations and forming a cascade state estimation scheme. Subsequently, we quantitatively and qualitatively assess the proposed scheme by comparing it to other estimation structures in terms of accuracy and robustness to disturbances, both in simulation and on an actual NAO robot walking outdoors over an inclined terrain. To facilitate further research endeavors, our implementation is offered as an open-source ROS/C++ package.

3.1 Aim and Contribution

Accurate 3D-CoM estimation is of vital importance for both walking pattern generation and real-time gait stabilization. In this chapter, we propose a nonlinear state estimation framework (illustrated in Fig. 3.1) for accurately estimating the 3D-CoM position, velocity and external forces acting on the CoM by effectively utilizing joint encoder, FSR, and IMU measurements. Starting from the Newton-Euler humanoid dynamics, we rigorously derive a nonlinear CoM estimator that uses as input the 3D-COP, the vertical GRF, and

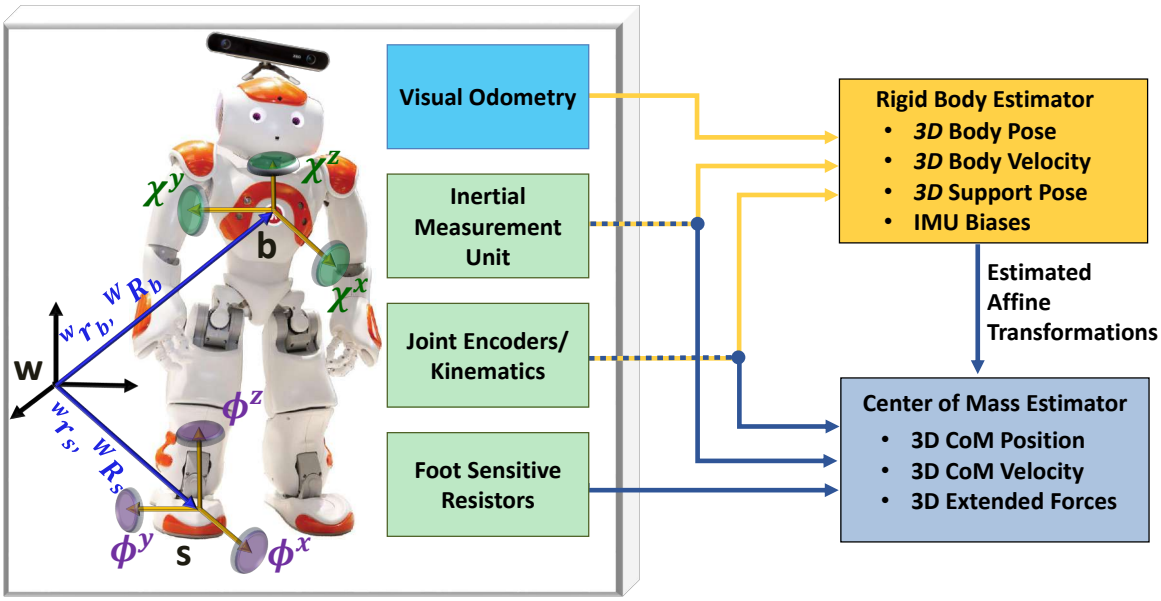

Figure 3.1: Cascade state estimation scheme consisting of a rigid body estimator and a CoM estimator.

the horizontal angular momentum rate. The output of the estimator is formulated as the 3D-CoM position along with the 3D-CoM acceleration. To the best of our knowledge, this is the first time that a CoM estimator explicitly considers the ground height and the angular momentum rate without relying on F/T sensors to yield, besides the 3D-CoM position and velocity, accurate 3D external force estimates. Contrasted to 3D-CoM estimation scheme presented in Chapter 2, the modeling errors in the acceleration level in this formulation represent exactly the external forces acting on the CoM and, furthermore, the angular momentum rate is taken into direct account. Thus, it is possible to provide more accurate estimates, when the motion is highly dynamic and the angular momentum rate is significant. In addition, this estimator can cope with cases of walking on uneven terrain, since the height of the ground is properly considered. As it is standard practice in CoM estimators, all measurements before fused are transformed from their local frames to the world frame. Therefore, by extending the rigid body estimator in [53], we provide the indispensable transformations that link the robot's body and support foot to a world frame, by fusing the onboard joint encoders, IMU and the pose obtained with visual odometry. Contrasted to [53], our approach differs in that: (a) the 3D-support foot position and orientation are properly considered, (b) kinematically computed 3D-relative support foot position and orientation are fused, (c) visual odometry measurements are considered, and (d) the linearizations for the aforementioned quantities are derived. In addition, contrasted

to [57], the proposed estimator: (a) maintains a robocentric state-space which improves the linearization accuracy and reduces drift [85], (b) incorporates visual odometry measurements, (c) considers only the support foot in the state which reduces the dimension of the filter by six, and (d) maintains rotational quantities directly as rotation matrices.

3.2 Center of Mass Estimation

In this section, we formally derive a nonlinear CoM state estimator and investigate its observability properties. In the following all quantities listed are in the world frame and the x, y, z superscripts indicate the corresponding vector coordinates. Consider the Newton-Euler equations of motion for a humanoid robot, where the ground contact forces $\boldsymbol{f}_i$ are explicitly separated from the external forces $\boldsymbol{f}_e$ acting on the CoM:

$$m(\ddot{\boldsymbol{c}} - \boldsymbol{g}) = \boldsymbol{f}_e + \sum_i \boldsymbol{f}_i \tag{3.1}$$

$$m\boldsymbol{c} \times (\ddot{\boldsymbol{c}} - \boldsymbol{g}) + \dot{\boldsymbol{L}} = \boldsymbol{c} \times \boldsymbol{f}_e + \sum_i \boldsymbol{s}_i \times \boldsymbol{f}_i \tag{3.2}$$

where $\boldsymbol{c}$ is the CoM position, $\ddot{\boldsymbol{c}}$ is the CoM acceleration, $\dot{\boldsymbol{L}}$ is the rate of angular momentum, m is the mass of the robot, and $\boldsymbol{g}$ is the gravity vector. Since $\boldsymbol{s}_i$ are the position of the contact points, the COP is defined as:

$$\boldsymbol{p} = \begin{bmatrix} \frac{\sum_i s_i^x f_i^z}{\sum_i f_i^z} & \frac{\sum_i s_i^y f_i^z}{\sum_i f_i^z} & s^z \end{bmatrix} \tag{3.3}$$

where we assume that in each foot, contact points are coplanar with respect to the local foot frame.

Then, by solving the first two equations of (3.2) for $\ddot{c}^x$ and $\ddot{c}^y$ while also considering (3.1), we get:

$$\ddot{c}^x = \frac{(c^x - p^x)(m(\ddot{c}^z - g^z) - f_e^z) - \dot{L}^y}{m(c^z - p^z)} + \frac{1}{m} f_e^x \tag{3.4}$$

$$\ddot{c}^y = \frac{(c^y - p^y)(m(\ddot{c}^z - g^z) - f_e^z) + \dot{L}^x}{m(c^z - p^z)} + \frac{1}{m} f_e^y \tag{3.5}$$

Examining the z component of (3.1) and introducing $\sum_i f_i^z = f_N$ as the vertical GRF, we get:

$$\ddot{c}^z = \frac{1}{m}(f_N + f_e^z) + g^z \tag{3.6}$$

By substituting (3.6) in (3.4) and (3.5), we readily obtain the nonlinear dynamics that

our CoM estimator is based on:

$$\ddot{c}^x = \frac{c^x - p^x}{m(c^z - p^z)} f_N - \frac{\dot{L}^y}{m(c^z - p^z)} + \frac{1}{m} f_e^x \tag{3.7}$$

$$\ddot{c}^y = \frac{c^y - p^y}{m(c^z - p^z)} f_N + \frac{\dot{L}^x}{m(c^z - p^z)} + \frac{1}{m} f_e^y \tag{3.8}$$

$$\ddot{c}^z = \frac{1}{m} \left(f_N + f_e^z \right) + g^z \tag{3.9}$$

3.2.1 CoM Estimator Process Model

For deriving the state-space needed in the EKF formulation, we assume a flying-wheel on the body with inertia $\boldsymbol{I}_b$. The latter is constantly computed based on the configuration of the limbs, to approximate the rate of angular momentum:

$$\dot{\boldsymbol{L}} = \boldsymbol{I}_b \, \dot{\boldsymbol{\omega}}_b + \boldsymbol{\omega}_b \times \boldsymbol{I}_b \, \boldsymbol{\omega}_b \tag{3.10}$$

where $\boldsymbol{\omega}_b$ is the gyro rate. Note that the second term in (3.10) accounts for the Coriolis and centrifugal effects. Subsequently, the following state vector is formulated:

$$\boldsymbol{x}_t^c = \left[c^x \; c^y \; c^z \; \dot{c}^x \; \dot{c}^y \; \dot{c}^z \; f_e^x \; f_e^y \; f_e^z \right]^\top$$

where the superscript c denotes the CoM estimator.

Furthermore, let the filter's input $\boldsymbol{u}_t^c$ be the location of the COP $\boldsymbol{p}$ in the 3D space with respect to the world frame, along with the vertical GRF f_N as measured by the FSRs. In addition, we compute the gyro acceleration $\dot{\boldsymbol{\omega}}_b$ by numerical differentiation of the IMU's gyro rate. Since numerical differentiation amplifies noise, we filter the gyro acceleration with a small window moving average filter to avoid introducing significant delays and phase shifts.

To this end, the input vector is:

$$\boldsymbol{u}_t^c = \left[p^x \; p^y \; p^z \; f_N \; \dot{L}^x \; \dot{L}^y \right]^\top \tag{3.11}$$

and the process model assumes the standard nonlinear form:

$$\dot{\boldsymbol{x}}_t^c = \boldsymbol{f}(\boldsymbol{x}_t^c, \boldsymbol{u}_t^c) + \boldsymbol{w}_t^c \tag{3.12}$$

where

$$\frac{d}{dt}\begin{bmatrix} c^x \\ c^y \\ c^z \\ \dot{c}^x \\ \dot{c}^y \\ \dot{c}^z \\ f_e^x \\ f_e^y \\ f_e^z \end{bmatrix} = \begin{bmatrix} \dot{c}^x \\ \dot{c}^y \\ \dot{c}^z \\ \frac{c^x-p^x}{m(c^z-p^z)} f_N - \frac{\dot{L}^y}{m(c^z-p^z)} + \frac{1}{m} f_e^x \\ \frac{c^y-p^y}{m(c^z-p^z)} f_N + \frac{\dot{L}^x}{m(c^z-p^z)} + \frac{1}{m} f_e^y \\ \frac{1}{m}(f_N + f_e^z) + g^z \\ 0 \\ 0 \\ 0 \end{bmatrix} + \boldsymbol{w}_t^c \tag{3.13}$$

with $\boldsymbol{w}_t^c$ the process additive noise. The linearization of the state-space is straightforward. The state Jacobian of the dynamics is:

$$\boldsymbol{F}_t = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{G}_t & \boldsymbol{0} & \boldsymbol{M}_t \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{3.14}$$

where

$$\boldsymbol{G}_t = \begin{bmatrix} \frac{f_N}{m(c^z-p^z)} & 0 & -\frac{(c^x-p^x)f_N - \dot{L}^y}{m(c^z-p^z)^2} \\ 0 & \frac{f_N}{m(c^z-p^z)} & -\frac{(c^y-p^y)f_N + \dot{L}^x}{m(c^z-p^z)^2} \\ 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{M}_t = \begin{bmatrix} \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{m} \end{bmatrix}$$

3.2.2 CoM Estimator Measurement Model

The measurements fused in the update step are the kinematically computed CoM position $\boldsymbol{c}^{\text{enc}}$ and the IMU CoM acceleration $\ddot{\boldsymbol{c}}^{\text{imu}}$, computed as in [86]. This approximation, as well as the approximation in (3.10), are valid as long as the actual CoM is located inside the same rigid link as the IMU, i.e. the body link. The latter has been proved valid in [87].

Accordingly, since the CoM acceleration is not part of the state, the measurement model is also nonlinear:

$$\boldsymbol{y}_t^c = \boldsymbol{h}(\boldsymbol{x}_t^c, \boldsymbol{u}_t^c) + \boldsymbol{n}_t^c \tag{3.15}$$

with

$$\boldsymbol{h}(\boldsymbol{x}_t^c, \boldsymbol{u}_t^c) = \begin{bmatrix} c^x \\ c^y \\ c^z \\ \frac{c^x - p^x}{m(c^z - p^z)} f_N - \frac{\dot{L}^y}{m(c^z - p^z)} + \frac{1}{m} f_e^x \\ \frac{c^y - p^y}{m(c^z - p^z)} f_N + \frac{\dot{L}^x}{m(c^z - p^z)} + \frac{1}{m} f_e^y \\ \frac{1}{m}(f_N + f_e^z) + g^z \end{bmatrix} \tag{3.16}$$

and $\boldsymbol{n}_t^c$ the additive Gaussian noise. The measurement model linearization is derived similarly as in section 3.2.1.

Notice, by using (3.7)-(3.9) both in the process and in the measurement model, the disturbance input noise correlates with the measurement noise. Still this has no effect on the estimation error itself but rather on the error covariance. Moreover, when the cross-correlation is zero, all expressions reduce to the EKF formulas. A formal proof is included in the appendix B section 9.4. In all our walking experiments, including ones of sufficient duration, we haven't noticed any degradation in the estimation accuracy of the error covariance. Thus, the cross-correlation noise ought to be insignificantly small.

3.2.3 Nonlinear Observability Analysis

In this section, we investigate the observability properties of the proposed CoM estimator in terms of the local nonlinear observability matrix. Following the approach in [88], that allows the nonlinear observability analysis to take into account output dynamics that depend explicitly on the input $\boldsymbol{u}$, we define the following coordinates $(h_1, \varphi_1^1, h_2, \varphi_2^1, h_3, \varphi_3^1, h_4, h_5, h_6)$, on the current operating point $({}^*\boldsymbol{x}_t^c, {}^*\boldsymbol{u}_t^c)$ where h_j is the $j-$th row of $\boldsymbol{h}({}^*\boldsymbol{x}_t^c, {}^*\boldsymbol{u}_t^c)$ and $\varphi_i^1 = L_f h_i$ is the Lie derivative of h_i in the direction of the vector field $\boldsymbol{f}({}^*\boldsymbol{x}_t^c, {}^*\boldsymbol{u}_t^c)$. Using these coordinates, we form the map [88]:

$$\boldsymbol{\Phi}({}^*\boldsymbol{x}_t^c, {}^*\boldsymbol{u}_t^c) = \begin{bmatrix} {}^*c^x \\ {}^*\dot{c}^x \\ {}^*c^y \\ {}^*\dot{c}^y \\ {}^*c^z \\ {}^*\dot{c}^z \\ \frac{{}^*c^x - {}^*p^x}{m({}^*c^z - {}^*p^z)} {}^*f_N - \frac{{}^*\dot{L}^y}{m({}^*c^z - {}^*p^z)} + \frac{1}{m} {}^*f_e^x \\ \frac{{}^*c^y - {}^*p^y}{m({}^*c^z - {}^*p^z)} {}^*f_N + \frac{{}^*\dot{L}^x}{m({}^*c^z - {}^*p^z)} + \frac{1}{m} {}^*f_e^y \\ \frac{1}{m}({}^*f_N + {}^*f_e^z) + g^z \end{bmatrix}$$

Subsequently, the components are re-ordered for convenience to form $\tilde{\boldsymbol{\Phi}}$. Computing the Jacobian with respect to ${}^*\boldsymbol{x}_t$, we get the local nonlinear observability matrix:

$$\boldsymbol{\mathcal{O}} = \frac{\partial \tilde{\boldsymbol{\Phi}}({}^*\boldsymbol{x}_t, {}^*\boldsymbol{u}_t)}{\partial {}^*\boldsymbol{x}_t} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ {}^*\boldsymbol{G}_t & \boldsymbol{0} & {}^*\boldsymbol{M}_t \end{bmatrix} \tag{3.17}$$

Ignoring the unrealistic case where $c^z = p^z$, meaningly the CoM lies exactly on the ground, we find that the nonlinear local observability matrix is full rank and cannot drop rank, since $\det\boldsymbol{\mathcal{O}} = \frac{1}{\mathrm{m}^3}$. Thus, the dynamics in (3.12), (3.15) are locally observable in all cases.

3.2.4 The need for Rigid Body Estimation

All measurements fused by the CoM estimator must be in an inertial frame of reference. Still, the latter are typically obtained in local frames, i.e. the kinematically computed CoM ${}^b\boldsymbol{c}^{\mathrm{enc}}$ is derived in the body frame and the measured by the FSR COP ${}^s\boldsymbol{p}^{\mathrm{fsr}}$ is in the support foot frame. Accordingly, they must be transformed to the world frame as:

$${}^w\boldsymbol{c}^{\mathrm{enc}} = {}^w\boldsymbol{r}_b + {}^w\boldsymbol{R}_b\,{}^b\boldsymbol{c}^{\mathrm{enc}} \tag{3.18}$$

$${}^w\boldsymbol{p}^{\mathrm{fsr}} = {}^w\boldsymbol{r}_s + {}^w\boldsymbol{R}_s\,{}^s\boldsymbol{p}^{\mathrm{fsr}} \tag{3.19}$$

with ${}^w\boldsymbol{r}_b$ and ${}^w\boldsymbol{r}_s$ the position of the body and support foot with respect to the world frame, ${}^w\boldsymbol{R}_b$ and ${}^w\boldsymbol{R}_s$ the corresponding orientations expressed as rotation matrices, as shown in Fig. 3.2. To this end, having reliable estimates of the body and support foot transformations is crucial in CoM estimation.

3.3 Rigid body Estimation

In [53] a rigid body state estimator based on Newton-Euler dynamics of a floating mass was presented. Since, the transformation linking the support foot to the world frame is mandatory to use quantities measured in the support foot frame such as the GRFs and the COP, we appropriately extend the process and measurement models to be able to estimate the following state vector:

$$\boldsymbol{x}_t^r = \begin{bmatrix} {}^b\boldsymbol{v}_b & {}^w\boldsymbol{R}_b & {}^w\boldsymbol{r}_b & {}^w\boldsymbol{R}_s & {}^w\boldsymbol{r}_s & \boldsymbol{b}_\omega & \boldsymbol{b}_a \end{bmatrix}^\top$$

where the superscript r denotes the rigid body estimator, ${}^b\boldsymbol{v}_b$ is the body's velocity, and $\boldsymbol{b}_\omega$, $\boldsymbol{b}_\alpha$ are the gyro and accelerometer biases, all expressed in the body local frame.

This EKF provides the necessary for CoM estimation rigid body transformations and accordingly preserves their affine properties. Hence, given Gaussian inputs the probabil-

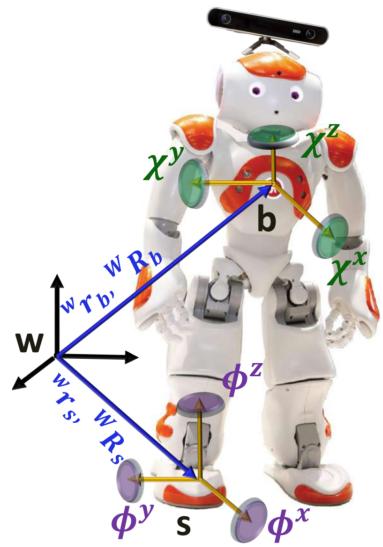

Figure 3.2: Illustration of frames needed for CoM estimation. Ellipses represent the orientation uncertainties in the corresponding local frame.

ity densities of the transformed output quantities remain Gaussians and thus the formed cascade estimation scheme does not give rise to inconsistencies during filtering.

3.3.1 Rigid Body Estimator Process Model

Let ${}^b\bar{\boldsymbol{\omega}}_b = {}^b\boldsymbol{\omega}_b^{\text{imu}} - \boldsymbol{b}_\omega$ and ${}^b\bar{\boldsymbol{\alpha}}_b = {}^b\boldsymbol{\alpha}_b^{\text{imu}} - \boldsymbol{b}_\alpha$, represent the IMU bias-removed gyro rate and linear acceleration, respectively. Then the nonlinear state-space takes the form:

$$
\begin{aligned}
{}^b\dot{\boldsymbol{v}}_b &= -({}^b\bar{\boldsymbol{\omega}}_b - \boldsymbol{w}_\omega) \times {}^b\boldsymbol{v}_b + {}^w\boldsymbol{R}_b^\top \boldsymbol{g} + {}^b\bar{\boldsymbol{\alpha}}_b - \boldsymbol{w}_a && (3.20)\\
{}^w\dot{\boldsymbol{R}}_b &= {}^w\boldsymbol{R}_b({}^b\bar{\boldsymbol{\omega}}_b - \boldsymbol{w}_\omega)_{[\times]} && (3.21)\\
{}^w\dot{\boldsymbol{r}}_b &= {}^w\boldsymbol{R}_b{}^b\boldsymbol{v}_b && (3.22)\\
{}^w\dot{\boldsymbol{R}}_s &= {}^w\boldsymbol{R}_s\boldsymbol{w}_{s[\times]} && (3.23)\\
{}^w\dot{\boldsymbol{r}}_s &= \boldsymbol{w}_{r_s} && (3.24)\\
\dot{\boldsymbol{b}}_\omega &= \boldsymbol{w}_{b_\omega} && (3.25)\\
\dot{\boldsymbol{b}}_\alpha &= \boldsymbol{w}_{b_\alpha} && (3.26)
\end{aligned}
$$

where $[\times]$ denotes the wedge operation and (3.23), (3.24) have been introduced to model the support foot orientation and position as random walks, since the foot in contact may or may not be stationary due to possible slippage. Furthermore, $\boldsymbol{w}_\omega$ and $\boldsymbol{w}_\alpha$ are the IMU noise vectors for the gyro rate and the linear acceleration, respectively, $\boldsymbol{w}_s$ and $\boldsymbol{w}_{r_s}$ are the

support foot orientation and position noises and $\boldsymbol{w}_{b_\omega}$, $\boldsymbol{w}_{b_\alpha}$ are the IMU bias noises.

To track the body's and support foot's orientation uncertainty we consider perturbation rotations in the corresponding local frames. Thus, if the true body and support rotation matrices are ${}^{w}\boldsymbol{R}_b$ and ${}^{w}\boldsymbol{R}_s$, then ${}^{w}\boldsymbol{R}_b = {}^{w}\hat{\boldsymbol{R}}_b \boldsymbol{e}^{\boldsymbol{\chi}_{[\times]}}$ and ${}^{w}\boldsymbol{R}_s = {}^{w}\hat{\boldsymbol{R}}_s \boldsymbol{e}^{\boldsymbol{\phi}_{[\times]}}$ where ${}^{w}\hat{\boldsymbol{R}}_b$, ${}^{w}\hat{\boldsymbol{R}}_s$ are the estimated rotation matrices and $\boldsymbol{\chi}$, $\boldsymbol{\phi}$ are the corresponding error exponential coordinates. An illustration of those quantities is given in Figure 3.2, where the black frame is the world frame, the yellow frames indicate the local body and support foot frames, while green and purple circles represent the corresponding orientation errors.

Subsequently, the linearization of (3.20) - (3.26) is derived as:

$$
\begin{aligned}
{}^{b}\boldsymbol{\delta}\dot{\boldsymbol{v}}_b =& -{}^{b}\bar{\boldsymbol{\omega}}_b \times {}^{b}\boldsymbol{\delta}\boldsymbol{v}_b + \left({}^{w}\boldsymbol{R}_b^{\top}\boldsymbol{g}\right) \times \boldsymbol{\chi} \\
& - {}^{b}\boldsymbol{v}_b \times (\boldsymbol{\delta}\boldsymbol{b}_\omega + \boldsymbol{w}_\omega) - \boldsymbol{\delta}\boldsymbol{b}_\alpha - \boldsymbol{w}_\alpha && (3.27)\\
\dot{\boldsymbol{\chi}} =& -{}^{b}\bar{\boldsymbol{\omega}}_b \times \boldsymbol{\chi} - \boldsymbol{\delta}\boldsymbol{b}_\omega - \boldsymbol{w}_\omega && (3.28)\\
{}^{w}\boldsymbol{\delta}\dot{\boldsymbol{r}}_b =& {}^{w}\boldsymbol{R}_b\, {}^{b}\boldsymbol{\delta}\boldsymbol{v}_b - {}^{w}\boldsymbol{R}_b \left({}^{b}\boldsymbol{v}_b \times \boldsymbol{\chi}\right) && (3.29)\\
\dot{\boldsymbol{\phi}} =& \boldsymbol{w}_s && (3.30)\\
\boldsymbol{\delta}\dot{\boldsymbol{r}}_s =& \boldsymbol{w}_{r_s} && (3.31)\\
\boldsymbol{\delta}\dot{\boldsymbol{b}}_\omega =& \boldsymbol{w}_{b_\omega} && (3.32)\\
\boldsymbol{\delta}\dot{\boldsymbol{b}}_\alpha =& \boldsymbol{w}_{b_\alpha} && (3.33)
\end{aligned}
$$

3.3.2 Rigid Body Estimator Measurement Model

The output model, formulated in [53], consists of the global body velocity using the robot's kinematics and the global body position and orientation using a LIDAR sensor and a Gaussian particle filter. To obtain the body velocity, the body position was computed using the filter's estimated orientation and the kinematically computed relative position of the support foot with respect to the body ${}^{b}\boldsymbol{r}_s^{\text{enc}}$ and then it was numerically differentiated. However, when using the estimated orientation (which is part of the state) for a measurement, correlation is induced to the filter. In addition, numerical differentiation commonly amplifies the noise and further filtering is needed.

Interestingly, it is possible to directly fuse ${}^{b}\boldsymbol{r}_s^{\text{enc}}$ since both the body and support foot position are available in our state. Moreover, the relative orientation ${}^{b}\boldsymbol{R}_s^{\text{enc}}$ must be also fused to render the support foot orientation observable:

$$
\begin{aligned}
{}^{b}\boldsymbol{r}_s^{\text{enc}} =& {}^{w}\boldsymbol{R}_b^{\top}\left({}^{w}\boldsymbol{r}_s - {}^{w}\boldsymbol{r}_b\right) + \boldsymbol{n}_{r_s} && (3.34)\\
{}^{b}\boldsymbol{R}_s^{\text{enc}} =& {}^{w}\boldsymbol{R}_b^{\top}\, {}^{w}\boldsymbol{R}_s \boldsymbol{e}^{\boldsymbol{n}_{r_{[\times]}}} && (3.35)
\end{aligned}
$$

with $\boldsymbol{n}_{r_s}$, $\boldsymbol{n}_r$ the kinematics measurement noise.

The previous measurements are typically available at a very fast rate. In this work, we

also employ measurements of the global head position and orientation by mounting an external camera on the head of the robot and using a visual odometry algorithm. The latter are then kinematically transformed to obtain the global body position and orientation and fused as:

$$
\begin{aligned}
{}^{w}\boldsymbol{r}_b^{\text{cam}} &= {}^{w}\boldsymbol{r}_b + \boldsymbol{n}_{\boldsymbol{r}_b} && (3.36)\\
{}^{w}\boldsymbol{R}_b^{\text{cam}} &= {}^{w}\boldsymbol{R}_b \boldsymbol{e}^{\boldsymbol{n}_{b_{[\times]}}} && (3.37)
\end{aligned}
$$

where $\boldsymbol{n}_{\boldsymbol{r}_b}$, $\boldsymbol{n}_b$ the camera measurement noise. This addition is essential, since leg odometry tends to drift and become inaccurate (see appendix B section 9.2 for leg odometry computation). Interestingly, this is also verified in the outdoors walking experiments presented in section 3.4.2.

For the linearization of the output model we consider the error exponential coordinates $\boldsymbol{\zeta}^{\text{enc}}$ and $\boldsymbol{\psi}^{\text{str}}$ related with ${}^{b}\boldsymbol{R}_s^{\text{enc}}$ and ${}^{w}\boldsymbol{R}_b^{\text{cam}}$, respectively. To this end, the linearization of (3.34) - (3.37) is given by:

$$
\begin{aligned}
{}^{b}\boldsymbol{\delta r}_s^{\text{enc}} &= {}^{w}\boldsymbol{R}_b^{\top}\left({}^{w}\boldsymbol{\delta r}_s - {}^{w}\boldsymbol{\delta r}_b\right)\\
&\quad + \left({}^{w}\boldsymbol{R}_b^{\top}\left({}^{w}\boldsymbol{r}_s - {}^{w}\boldsymbol{r}_b\right)\right) \times \boldsymbol{\chi} + \boldsymbol{n}_{\boldsymbol{r}_s} && (3.38)\\
\boldsymbol{\zeta}^{\text{enc}} &= -\left({}^{w}\boldsymbol{R}_s^{\top}{}^{w}\boldsymbol{R}_b\right)\boldsymbol{\chi} + \boldsymbol{\phi} + \boldsymbol{n}_{\boldsymbol{r}} && (3.39)\\
{}^{w}\boldsymbol{\delta r}_b^{\text{cam}} &= {}^{w}\boldsymbol{\delta r}_b + \boldsymbol{n}_{\boldsymbol{r}_b} && (3.40)\\
\boldsymbol{\psi}^{\text{cam}} &= \boldsymbol{\chi} + \boldsymbol{n}_b && (3.41)
\end{aligned}
$$

Formal proofs of the linearized process and output models are included in the appendix B section 9.5.

3.4 Results

The proposed framework has been implemented and experimentally validated. In the next section we outline quantitative, simulation-based results, that demonstrate the accuracy and robustness of the proposed estimator in simulated gaits over uneven terrain. Subsequently, we present results on a NAO robot, and demonstrate accurate external force estimation and how drift affects the CoM estimation, highlighting the significance of the proposed cascade scheme. Given that disturbances tend to be sudden and discrete events, we employ in all experiments high process noise in order to facilitate fast convergence of the estimated external forces.

3.4.1 Simulation Experiments

Humanoid Robot Walking over Rough Terrain

In order to obtain quantitative assessment results, we simulated a humanoid robot walking over uneven terrain, while our nonlinear CoM estimator is employed for feedback. The proposed CoM estimator, termed as EKF1, is contrasted to the nonlinear estimator in [47], termed as EKF2, and to a Linear KF (LKF) variant of [86], which is the only linear scheme fusing CoM acceleration. The latter estimates a CoM offset instead of the external forces, thus the offsets are transformed to forces as:

$$f_e^{x,y} = m\frac{g}{h_c}c_{\text{offset}}^{x,y} \tag{3.42}$$

where h_c is the nomimal CoM height. The selection of EKF2 and LKF schemes for comparison is due to the fact that EKF2 has been shown to be an accurate 3D-CoM estimator [47] and LKF is broadly utilized in the literature [86].

For all employed filters ideal base/support state estimation was assumed and the same noise covariances $\boldsymbol{Q}$ and $\boldsymbol{R}$ are used. The 3D-step positions are computed based on the terrain's shape while the motion generation is based on the DCM $\boldsymbol{\xi}$ with continuous Double Support (DS) phases [89]. The mass m and inertia $I_b^{xx,yy}$ of the robot along with h_c and step time T_s are shown in Table 3.1.

In this experiment, illustrated in Figure 3.3, the robot stands up, initializes its posture by taking two steps in place and starts to walk. During the third step and at $t = 6s$, a disturbance in the x axis of $2200N$ is introduced. After recovering within a step, another push happens in the y axis with intensity of $1500N$. Subsequently, in the following step the robot is perturbed in both x and y axes with $1800N$ and $1600N$, respectively. Due to this last push, early swing leg landing occurs causing a disturbance of approximately $1000N$ in the vertical axis. Finally, the robot manages to walk down the terrain unperturbed.

Figure 3.4 shows both the 3D-CoM position (top) and velocity (bottom) as estimated by the employed estimators contrasted to the ground-truth trajectories. In addition, Figure 3.5 illustrates the corresponding 3D-DCM position (top) and velocity (bottom) trajectories computed as in [47]. We observe that the proposed CoM estimator yields more accurate estimates, which is due to the fact that the ground height in the denominators of (3.7)

Table 3.1: Simulation Parameters.

$m\ (kg)$	$I_b^{xx}\ (kgm^2)$	$I_b^{yy}\ (kgm^2)$	$h_c(m)$	$T_s\ (s)$
156.5	0.33	0.30	0.812	1

	$\boldsymbol{c}$	$\dot{\boldsymbol{c}}$	$\boldsymbol{f}_e$	$\boldsymbol{c}_{\text{offset}}$		$\boldsymbol{c}^{\text{enc}}$	$\ddot{\boldsymbol{c}}^{\text{imu}}$
$\boldsymbol{Q}$	$1e^{-8}$	$1e^{-4}$	$5e^{1}$	$1e^{-8}$	$\boldsymbol{R}$	$1e^{-5}$	$4e^{-2}$

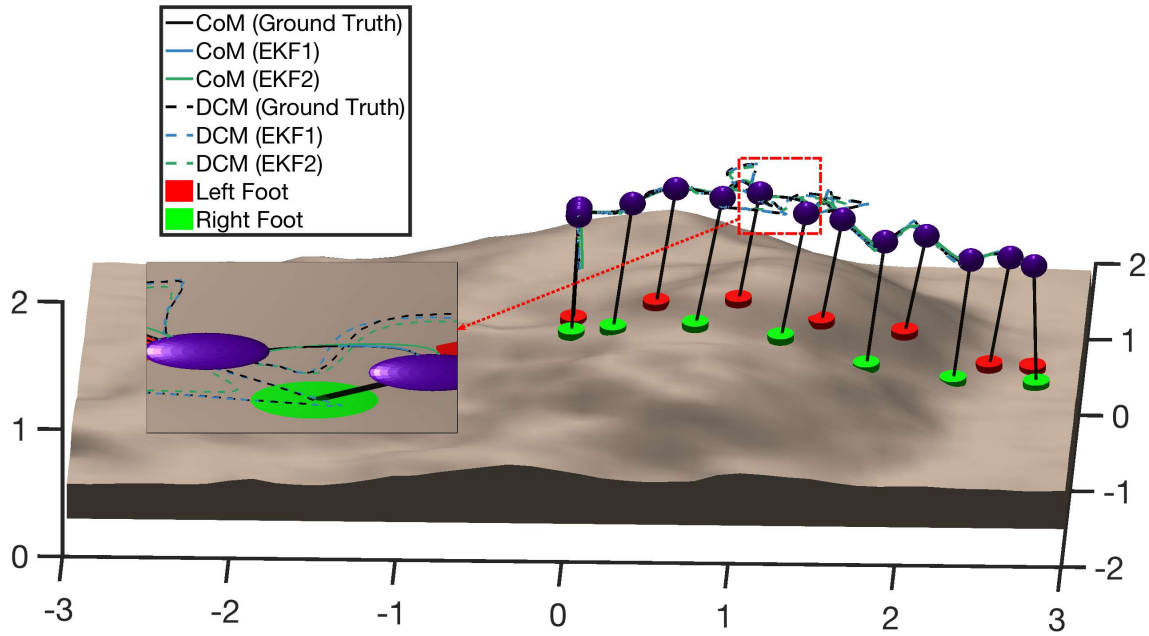

Figure 3.3: CoM/DCM trajectories in the 3D space, black / black–dotted lines indicate the ground truth trajectories, blue / blue–dotted lines the EKF1 estimated trajectories, green / green–dotted lines the EKF2 estimated trajectories and red / green the left and right support foot respectively. LKF yields no estimates since the z-axis is neglected.

and (3.8) along with the angular momentum rates translate to modeling errors, yielding inaccuracies for all the estimated quantities of EKF2 and LKF, while in the EKF1 case they are directly considered. This is also evident in Figure 3.6 illustrating the external forces, where as seen strong pushes, e.g. in x-axis, cause the robot to rotate about the y-axis, generating angular momentum and false appearing as external forces for EKF2 and LKF in that axis.

Moreover, EKF1 and EKF2, as expected, yield similar response in the z-axis since they are both based on (3.9). On the other hand, LKF yields no estimates since it assumes that CoM lies on a constant height plane.

Based on the above, in order to demonstrate the accuracy of the proposed CoM estimator, we conducted 100.000 simulations of 12 random omni-directional steps each. In every run, random disturbances varying in magnitude from $1-1.5$, $0.5-1$, and $0.25-0.5$ times the weight of the robot in the x, y and z axes respectively, were introduced at random time instances during the gait. Figure 3.7 illustrates the Root Mean Square Error (RMSE) from the ground truth trajectories during the perturbation periods for each estimator employed. The external forces are scaled by 10^{-3} for clarity. As evident, we gained a significant boost

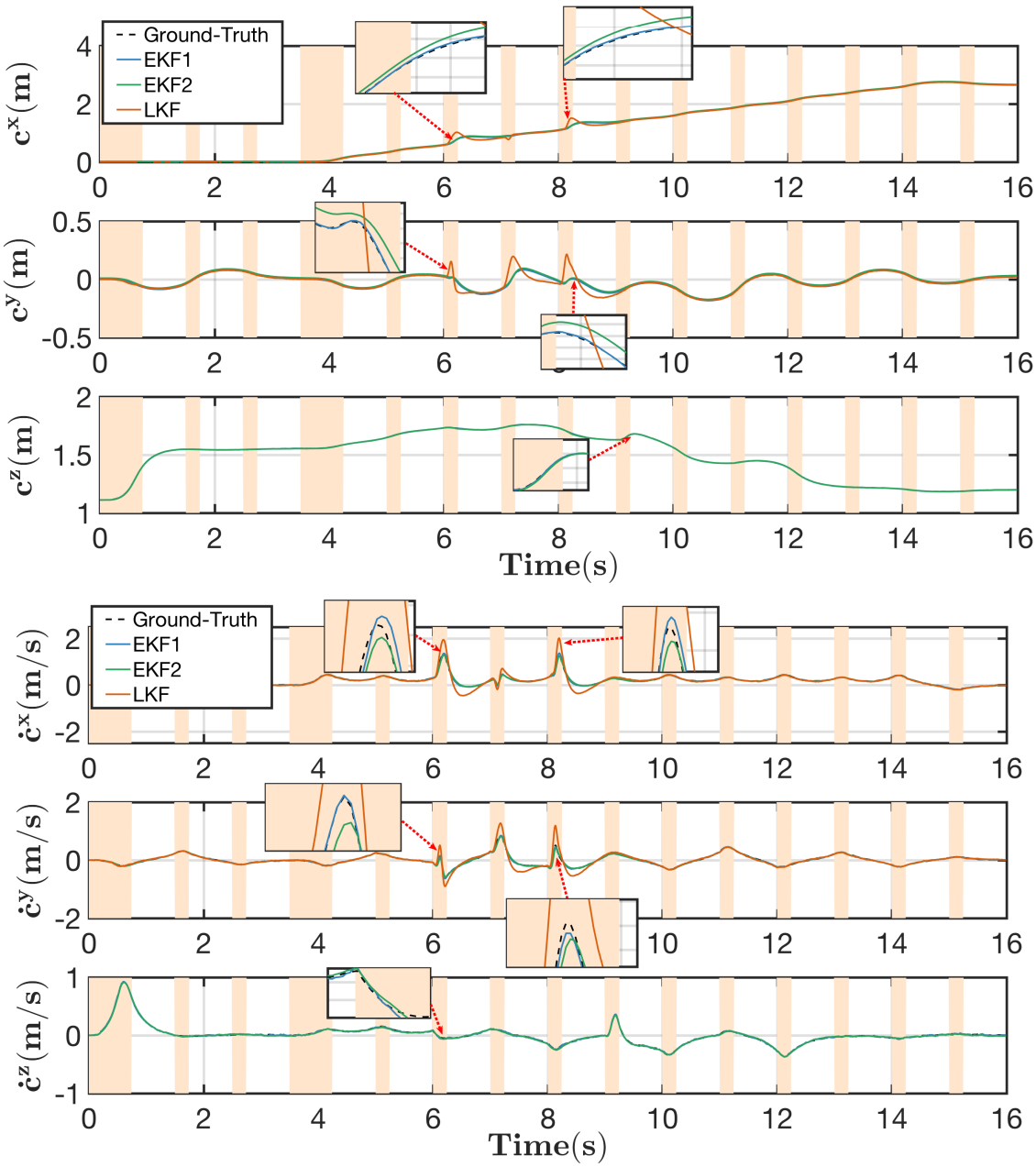

Figure 3.4: **Top:** 3D-CoM trajectories, **Bottom:** 3D-CoM velocities, light beige regions indicate the DS phases, black dotted lines indicate the ground truth trajectories, blue lines, green lines, and orange lines are the estimated trajectories by EKF1, EKF2, and LKF respectively.

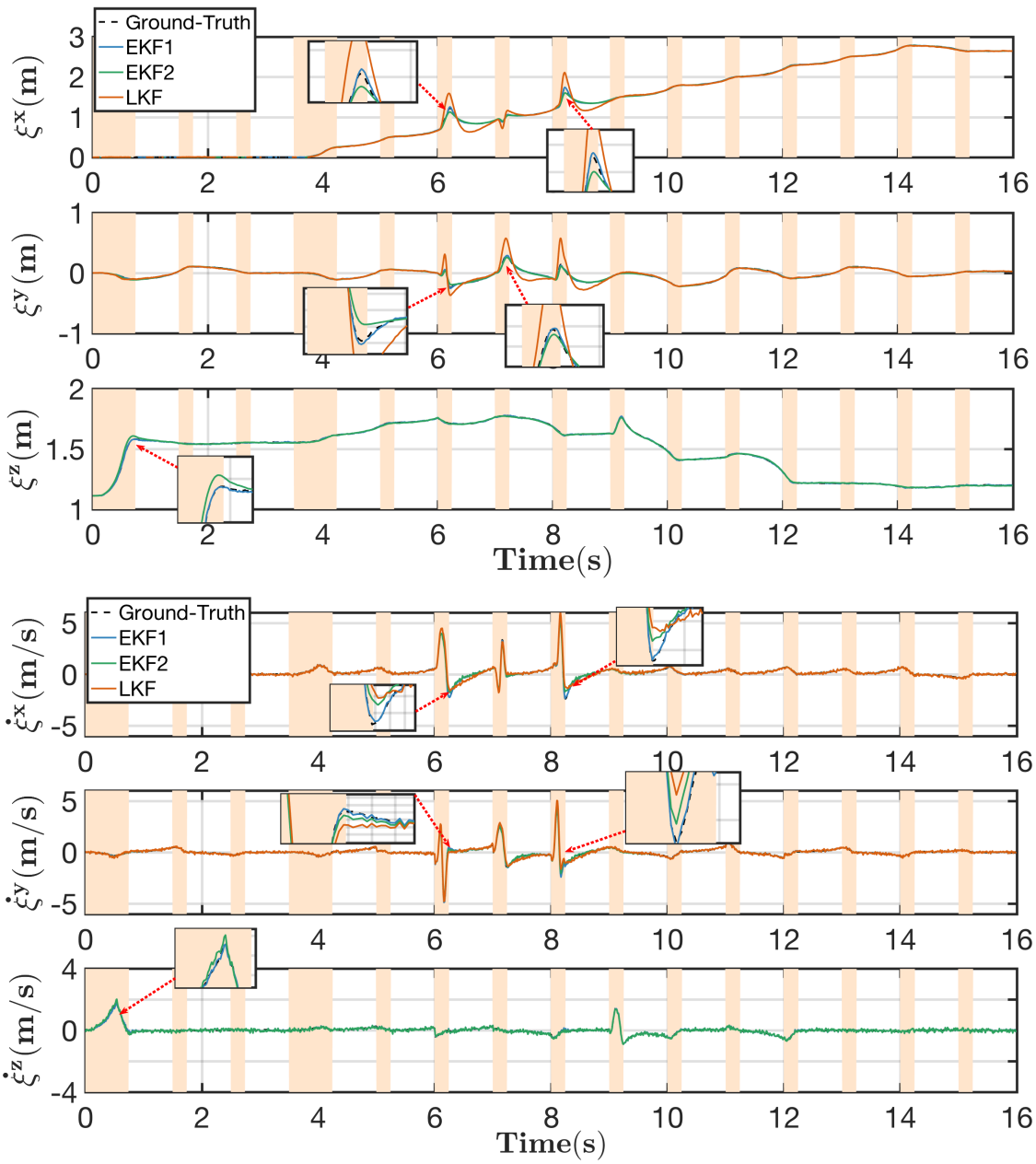

Figure 3.5: **Top:** 3D-DCM trajectories, **Bottom:** 3D-DCM velocities, light beige regions indicate the DS phases, black dotted lines indicate the ground truth trajectories, blue lines, green lines, and orange lines are the estimated trajectories by EKF1, EKF2, and LKF respectively.

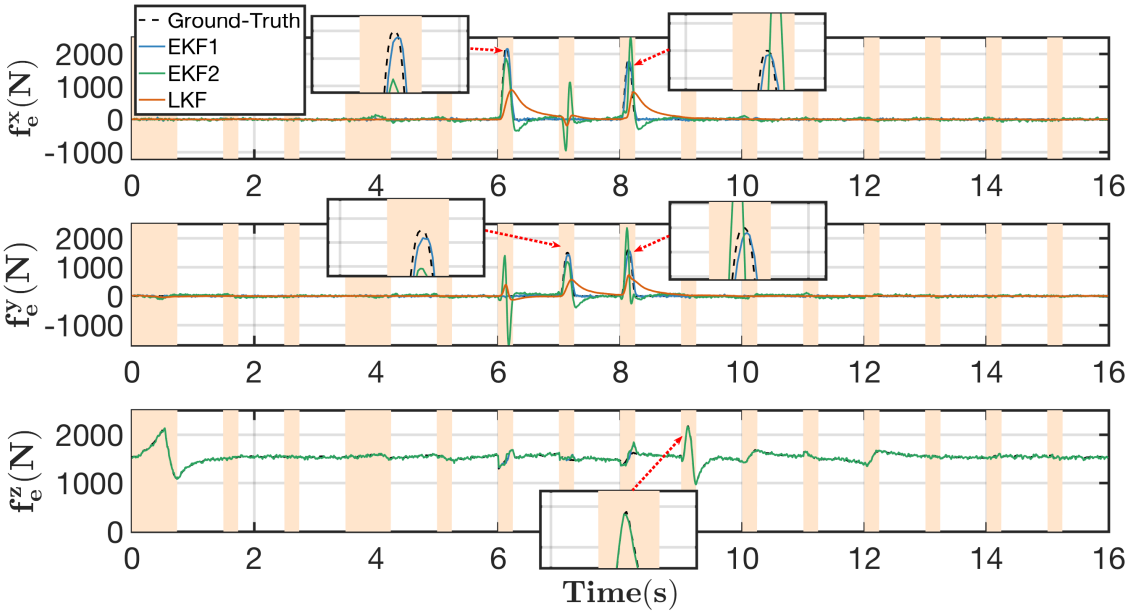

Figure 3.6: 3D-External forces, light beige regions indicate the DS phases, black dotted lines indicate the ground truth trajectories, blue lines, green lines, and orange lines are the estimated trajectories by EKF1, EKF2, and LKF respectively.

in accuracy for all quantities of interest in the x and y axes, especially in external forces, for only 14.32% extra computational cost.

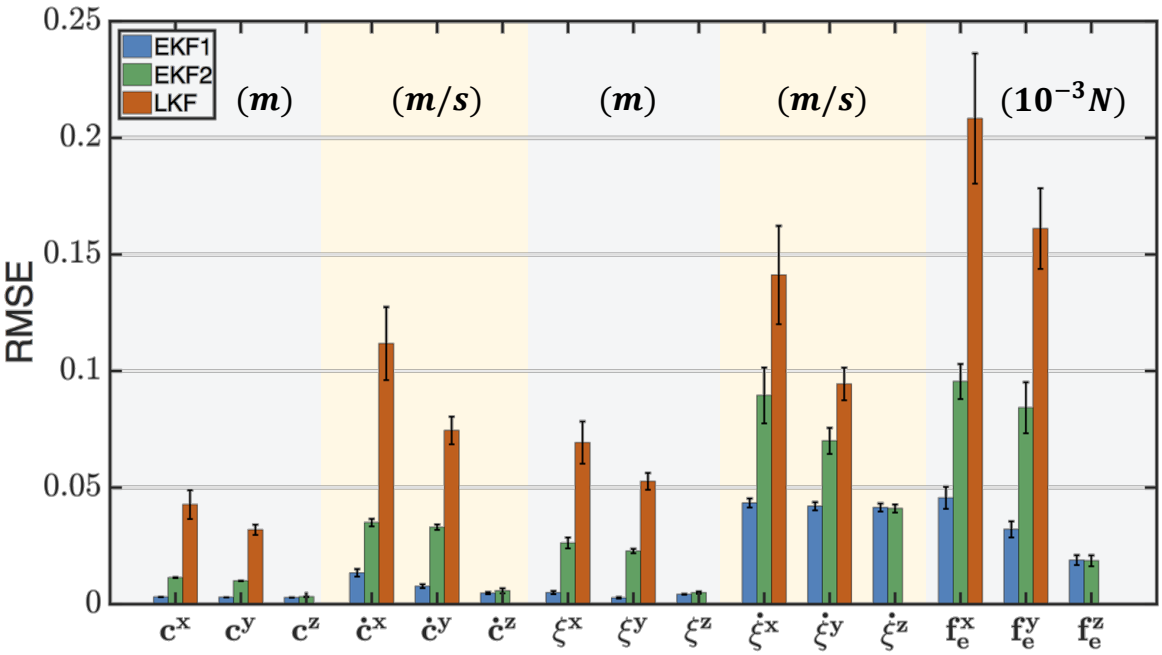

Figure 3.7: RMSE for CoM, DCM, and external forces during perturbation periods for 100.000 simulations; blue bars indicate the EKF1s error, green bars the EKF2s error, orange bars the LKFs error, and black lines the standard deviation from the corresponding mean values.

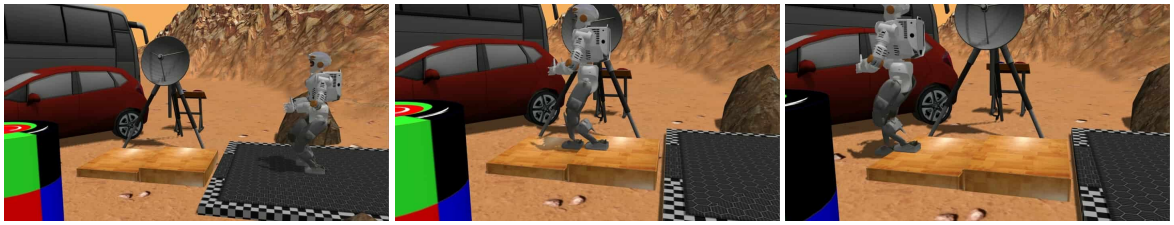

Figure 3.8: Valkyrie walking over uneven/rough terrain in Gazebo.

Valkyrie Humanoid Walking over Uneven Terrain

Next, we employ the proposed cascade framework in Gazebo with NASA's Valkyrie humanoid using our ROS open-source implementation running in real-time every $2ms$, where also the parameters used for this experiment are listed in the Valkyrie configuration file [90]. For walking over the uneven terrain, as evident in Figure 3.8, we utilized the control framework in [63]. The IMU measurments are available at $1kHz$ while the joint encoders and FSR measurements are obtained at $500Hz$. Furthermore, to compute the visual pose fused in our filter, we used the Semi-Direct Visual Odometry (SVO) [91] with the multisense stereo running at $40Hz$. In Figure 3.9 the 3D-Body position and velocity are illustrated. Notice the kinematically computed trajectories inevitably drift as the robot continuously walks, whereas, our rigid-body estimator, termed as EKF1, yielded accurate estimates for all quantities with respect to the ground-truth trajectories. Specifically, the RMSE for the body position were $0.0034m$, $0.0036m$ and $0.001m$, and for the body velocity $0.0139m/s$, $0.0159m/s$ and $0.0128m/s$ for the x, y and z axes, respectively.

Since Valkyrie is employed with 6D-F/T sensors in the feet (as opposed to the simulated robot in section 3.4.1 and NAO in Section 3.4.2), we compare the proposed CoM estimator (termed as EKF1 for simplicity) to the Momentum Estimator (ME) with external wrenches [46]. For both filters the same base/support information and noise covariances for the measurements in common are used, whereas the torque and external torque covariances were fine tuned in the ME case. Figure 3.10 illustrates the 3D-CoM position and velocities as estimated by each method. As evident, for the 3D-CoM position both estimators yielded the same response, while small differences arose in the estimated 3D-CoM velocities. Table 3.2 summarizes the RMSE of all estimated quantities for both filters. In this static, low pace gait, Valkyrie experienced mostly co-planar contacts, thus the proposed CoM estimator yielded very accurate estimates. Nevertheless, we expect ME to provide more accurate estimates in the general case where the robot exhibits non co-planar contacts, but at the cost of employing 6D-F/T sensors at the robot's end-effectors.

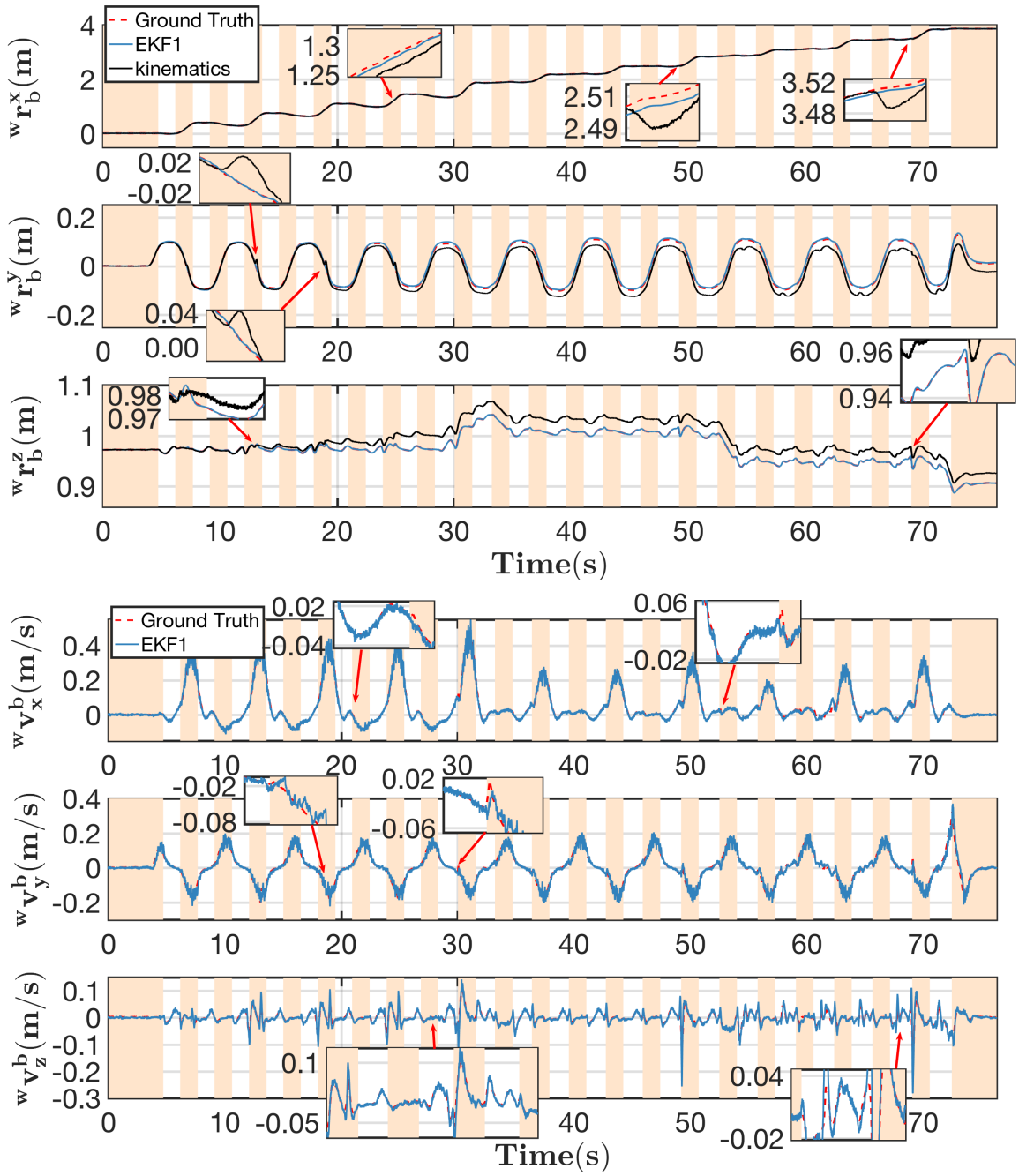

Figure 3.9: **Top:** 3D-Body trajectories, **Bottom:** 3D-Body velocities, light beige regions indicate the DS phases, black dotted lines the ground truth trajectories, blue lines the estimated trajectories by EKF1 and black lines the kinematically computed trajectories.

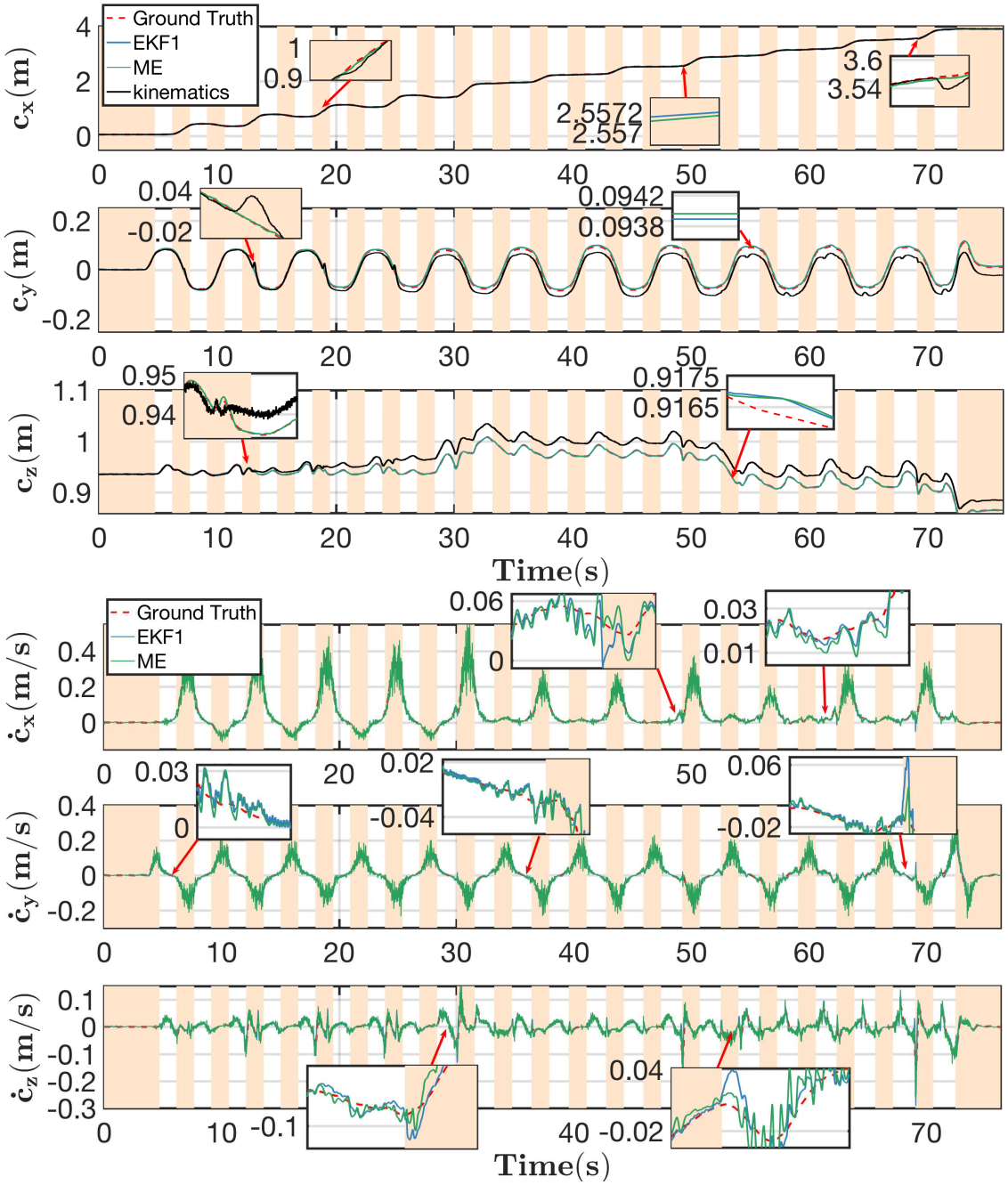

Figure 3.10: **Top:** 3D-CoM trajectories, **Bottom:** 3D-CoM velocities, light beige regions indicate the DS phases, blue and green lines are the estimated trajectories by EKF1 and ME.

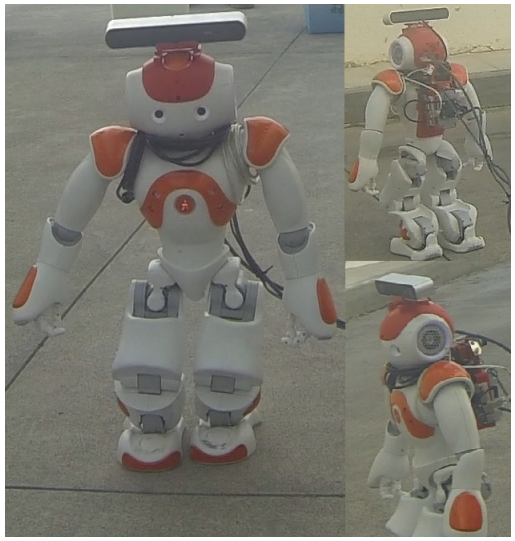

Figure 3.11: NAO humanoid walking over a pavement.

3.4.2 Real Robot Experiments

The proposed cascade framework was further implemented on a NAO v4.0 humanoid robot (Figure 3.11, running in real-time every $10ms$. The joint encoder, IMU and FSRs measurements needed by the scheme are available at a $100Hz$ rate. For obtaining the pose, we used SVO with a ZED stereo camera, running on an Nvidia Jetson TX2 module, communicating with NAO through ethernet with a TCP/IP server. The latter was available to NAO at an average rate of $40Hz$. The estimation parameters used can be found in the NAO configuration file in [90].

A first result regards estimation of the external forces, where the robot was disturbed and the pushes were accurately measured with an Alluris force gauge. The NAO robot was commanded to stand up, initialize its posture by making two steps, and then stand still. As we can see in Figure 3.12, from $6s$ to $13s$ where the NAO robot is unperturbed in the z-axis the external force counter balance the false measurement from the FSR for the total weight so that the resultant force $(f_N + f_e^z)$ yields the mass of the robot which is approximately $5.19kg$. Moreover, a disturbance in the x-axis is performed at $13s$ and settles at $16.6s$. This disturbance was measured to have a peak magnitude of $5.96N$, as also estimated by f_e^x. Finally, a constant lateral disturbance was enforced at $21.4s$ until $25.3s$ with peak at $11.64N$

Table 3.2: RMSE of Estimated Quantities.

	$c_x(m)$	$c_y(m)$	$c_z(m)$	$\dot{c}_x(\frac{m}{s})$	$\dot{c}_y(\frac{m}{s})$	$\dot{c}_z(\frac{m}{s})$
EKF1	0.0036	0.0037	0.0011	0.0179	0.0174	0.0123
ME	0.0036	0.0037	0.0011	0.0205	0.0212	0.0099

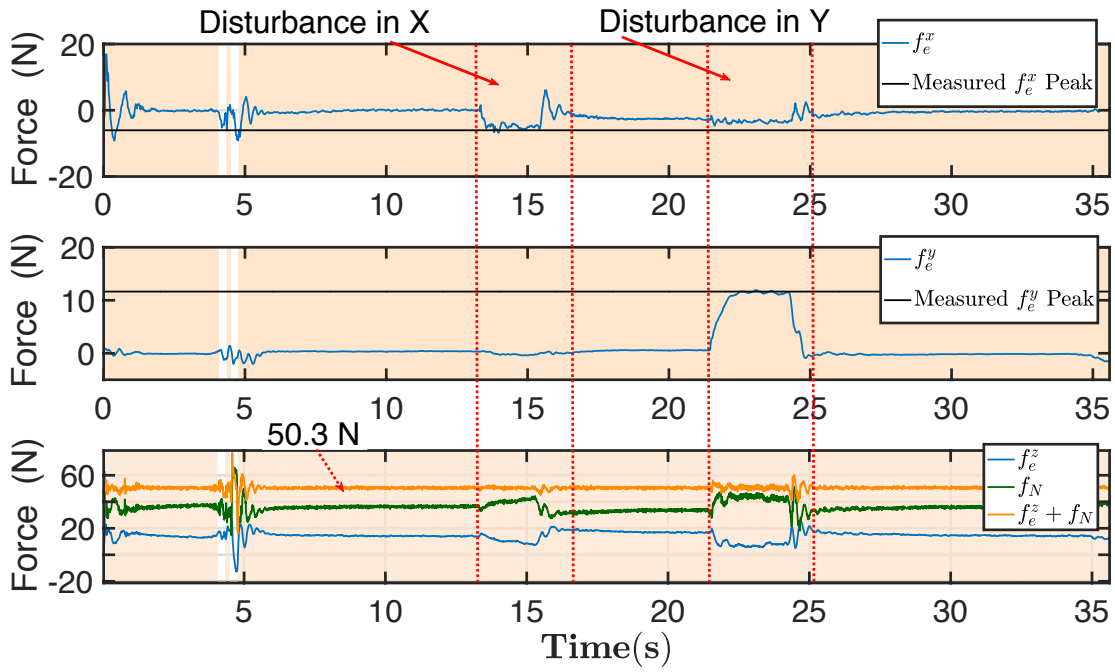

Figure 3.12: 3D-External forces, light beige regions correspond to the DS phases, blue lines indicate the estimated external forces by the proposed CoM estimator, orange is the vertical resultant force $f_e^z + f_N$, green is the FSR measured vertical GRF f_N, and black are the measured external force peaks.

making NAO tilt; again our estimator yielded an accurate estimate f_e^y.

To fair contrast the proposed cascade scheme, we constructed another serial state estimation scheme, based on the rigid body estimator of [53] and EKF2; for simplicity this is termed EKF2 in the sequel. Since a LIDAR sensor was not available, we used the camera pose for the global body position and orientation measurement. Furthermore, to remove the correlation explained in section 3.3.2, we computed the body velocity in the world frame using kinematics.

In addition, the transformation of the support foot with respect to the world frame was computed using the kinematic relative transformation from the body to the support and the estimated body to world transformation, since it cannot be estimated directly, as in our approach.

Subsequently, we let our robot walk outdoors on a challenging inclined terrain where the slope in the forward and in the lateral directions was $16°$ and $5°$, respectively. To accurately measure the final pose, since in outdoor environments ground truth data are not available, we used both conventional measuring tools and digital laser rangefinders to measure the final position and orientation (termed as Ground-Truth) at the end of the gait.

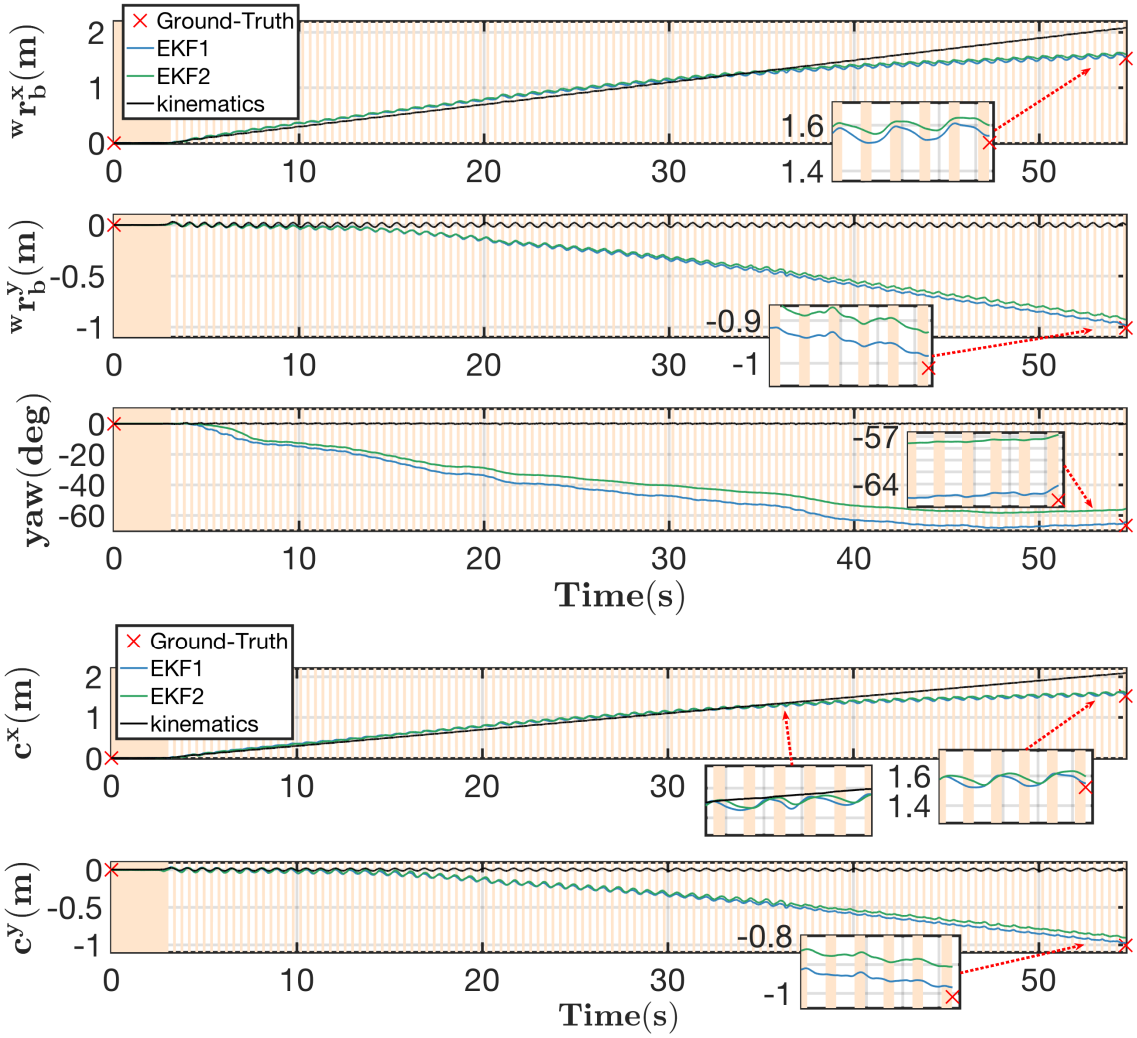

Figure 3.13: **Top:** 2D-Body pose trajectory, **Bottom:** 2D-CoM trajectory, light beige regions indicate the DS phases, $\times$ the ground truth positions, blue lines and green lines are the estimated trajectories with EKF1 and EKF2 respectively, and black lines are the kinematically computed trajectories.

In order to observe the drift and how drift can affect CoM estimation, we ordered NAO to walk straight up the inclined road. Figure 3.13 (top) shows the 2D body pose as estimated by the employed schemes and as computed using the kinematics. Both estimators yielded pretty similar results while walking straight where the drift was negligible. Small differences arise from the fact that in the proposed cascade scheme the support foot dynamics also work as constraints for respecting the robot kinematic chains. Nevertheless, when the robot started to drift, EKF2 accuracy started to degrade, since the kinematically computed

body velocity in the world frame is fused. EKF2 final pose error was $7.56cm$ and $8.43cm$ in the x and y-axes and $10.06°$ for the body yaw angle, while for the EKF1 it was $3.06cm$ and $2.88cm$ in the x and y-axes and $2.81°$ in the yaw angle. Notice, the kinematically computed odometry was completely off, since it shows that the robot had actually performed a straight gait. Accordingly, Figure 3.13 (bottom) shows that the same degradation in accuracy is inevitably inherited in the CoM estimation, demonstrating one more time that accurate rigid body estimation is vital to CoM estimation. In addition, we note that EKF1 yields a more oscillatory response, which is expected since when walking over inclined terrains early swing leg landing commonly occurs causing the robot to rotate and producing angular momentum, which is not considered in EKF2. All the presented experiments can be visualized in high resolution at `https://goo.gl/7kbcuf`.

3.5 Conclusion

In this chapter we proposed a novel cascade estimation scheme that fuses IMU, joint encoders, FSR and visual input to provide with accurate estimates of important quantities in humanoid planning and control.

After implementing the proposed scheme both in simulation and on a real NAO robot, we demonstrated its accuracy, robustness to disturbances and efficacy in realistic scenarios. Given that the proposed cascade scheme is based on generic/simplified dynamics, it is readily amenable to generalization to other humanoids. To this end, we released SEROW [90], an open-source ROS package to reinforce robotic research endeavors.

Nevertheless, state-of-the-art Visual Odometry (VO) or LIDAR Odometry (LO) algorithms implicitly assume that the world the robot exists in is static. To this end, the accuracy of the latter approaches significantly degrades in dynamic envinronments i.e. daily human environments, giving rise to outlier measurements. In the next chapter, we will propose a robust base estimator that automatically detects and discards outlier VO and LO measurements.

Chapter 4
Outlier-Robust State Estimation

The worthwhile problems are the ones you can really solve or help solve, the ones you can really contribute something to. ... No problem is too small or too trivial if we can really do something about it.

Richard Phillips Feynman (1918–1988)

Contemporary humanoids are equipped with visual and LIDAR sensors that are effectively utilized for Visual Odometry (VO) and LIDAR Odometry (LO). Unfortunately, such measurements commonly suffer from outliers in a dynamic environment, since frequently it is assumed that only the robot is in motion and the world is static. To this end, robust state estimation schemes are mandatory in order for humanoids to symbiotically co-exist with humans in their daily dynamic environments. In this chapter, the robust Gaussian Error-State Kalman Filter for humanoid robot locomotion is presented. The introduced method automatically detects and rejects outliers without relying on any prior knowledge on measurement distributions or finely tuned thresholds. Subsequently, the proposed method is quantitatively and qualitatively assessed in realistic conditions with the full-size humanoid robot WALK-MAN v2.0 and the mini-size humanoid robot NAO to demonstrate its accuracy and robustness when outlier VO/LO measurements are present. Finally, in order to reinforce further research endeavours, our implementation is released as an open-source ROS/C++ package.

4.1 Aim and Contribution

Modern humanoids are commonly employed with cameras and LIDAR sensors to reinforce their perception in unstructured environments. Based on consecutive camera frames one can derive the camera's egomotion with respect to the environment and directly relate it to the robot's motion. In literature this is known as Visual Odometry (VO). Prominent approaches rely on sparse [92] or semi-dense [91] schemes to facilitate real-time execu-

Figure 4.1: Illustration of frames used in base estimation on the 29DoF WALK-MAN v2.0 humanoid robot: w corresponds to the inertial world frame, b is the base frame, and blue ellipses indicate the orientation uncertainty.

tion. Similarly, based on sequential LIDAR scans it is straightforward to match the beams and compute the LIDAR Odometry (LO) [93, 94]. Both approaches are advantageous in the sense that they are unaffected by slippage in uneven/rough terrain when contrasted to the kinematically computed leg odometry. However, in all aforementioned schemes, the world is assumed to be mostly static and only the robot is in motion, e.g. the static world assumption. Presumably, this is not the case in human daily environments, due to humans moving along with the robots and/or changing the scene. Hence, the static world assumption is frequently violated. To this end, in order for humanoids to co-exist with humans in a dynamically changing environment it is mandatory to robustify their odometry estimates. Interestingly, in [74] a base estimator with outlier detection for quadruped locomotion was presented. The authors of [74] utilized a probabilistic threshold to quantify weather a measurement is an outlier or not before fusion. Nevertheless, this raises two important questions: a) *how can this threshold be determined in advance* and b) *does this threshold depend on the conditions at hand?* Other works [95], not in the scope of base estimation, assumed that the measurements follow a Student-t distribution. Again the obvious question arises whether this is a valid assumption in the case of VO/LO measurements.

In this chapter, we propose a novel formulation of the Error-State Kalman Filter (ESKF) which is robust to outlier VO/LO measurements that can commonly occur in humanoid walking in dynamic human environments. The contribution to the state-of-the-art is as

follows:

- The Robust Gaussian ESKF (RGESKF) is mathematically established based on [53,96]. More specifically, we present an analytical solution for the general nonlinear Gaussian formulation for outlier detection of [96]. The latter results in a computationally efficient implementation that accomplishes real-time execution.

- The above method does not rely on prior assumptions regarding the measurements probability distributions [95] neither thresholding [74] for the imminent outlier detection.

- We quantitatively and qualitatively assess the proposed method and demonstrate its accuracy and robustness in real-world conditions with two robots, the full-size humanoid WALK-MAN v2.0 [97], and a mini-size NAO humanoid.

- Since this framework relies on sensing that is commonly available on contemporary humanoids and furthermore, is based on generic nonlinear dynamics, we release an open-source ROS/C++ implementation [90] to reinforce further research endeavours.

4.2 Base Estimation Revisited

Kuindersma et al. [53], presented a base estimator with Newton-Euler dynamics of a floating mass that is effectively used in humanoid walking. At time t, the state to be estimated is:

$$\boldsymbol{x}_t = \begin{bmatrix} {}^b\boldsymbol{v}_b & {}^w\boldsymbol{R}_b & {}^w\boldsymbol{p}_b & \boldsymbol{b}_\omega & \boldsymbol{b}_a \end{bmatrix}^\top$$

where ${}^w\boldsymbol{p}_b$, ${}^w\boldsymbol{R}_b$ denote the base position and rotation with respect to the world frame w, ${}^b\boldsymbol{v}_b$ is the linear velocity, and $\boldsymbol{b}_\omega$, $\boldsymbol{b}_\alpha$ are the gyro and accelerometer biases, in the base frame b. However, ${}^w\boldsymbol{R}_b$ is an overparametrization of the base's orientation. To this end, to track the orientation uncertainty we consider perturbation rotations in the base frame. Thus, if the true base rotation matrix is ${}^w\boldsymbol{R}_b$ then ${}^w\boldsymbol{R}_b = {}^w\hat{\boldsymbol{R}}_b e^{\chi_{[\times]}}$ where ${}^w\hat{\boldsymbol{R}}_b$, is the estimated rotation matrix and χ denotes the perturbation exponential coordinates. For clarity all aforementioned quantities are depicted in Figure 4.1.

4.2.1 Process Model

In order to properly define the nonlinear dynamics $\boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{w}_t)$, let ${}^b\bar{\boldsymbol{\omega}}_b = {}^b\boldsymbol{\omega}_b^{\text{imu}} - \boldsymbol{b}_\omega$ and ${}^b\bar{\boldsymbol{\alpha}}_b = {}^b\boldsymbol{\alpha}_b^{\text{imu}} - \boldsymbol{b}_\alpha$, be the IMU bias-compensated gyro rate and linear acceleration, respectively, then:

$$\dot{\boldsymbol{x}}_t = \overbrace{\begin{bmatrix} -({}^b\bar{\boldsymbol{\omega}}_b - \boldsymbol{w}_{\boldsymbol{\omega}}) \times {}^b\boldsymbol{v}_b + {}^w\boldsymbol{R}_b^\top \boldsymbol{g} + {}^b\bar{\boldsymbol{\alpha}}_b - \boldsymbol{w}_{\boldsymbol{a}} \\ {}^w\boldsymbol{R}_b({}^b\bar{\boldsymbol{\omega}}_b - \boldsymbol{w}_{\boldsymbol{\omega}})_{[\times]} \\ {}^w\boldsymbol{R}_b{}^b\boldsymbol{v}_b \\ \boldsymbol{w}_{\boldsymbol{b}_{\boldsymbol{\omega}}} \\ \boldsymbol{w}_{\boldsymbol{b}_{\boldsymbol{\alpha}}} \end{bmatrix}}^{\boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{w}_t)} \tag{4.1}$$

where $\boldsymbol{u}_t = \begin{bmatrix} {}^b\boldsymbol{\omega}_b^{\text{imu}} & {}^b\boldsymbol{\alpha}_b^{\text{imu}} \end{bmatrix}$ is the input vector, $\boldsymbol{g}$ is the gravity vector, $\boldsymbol{w}_t = \begin{bmatrix} \boldsymbol{w}_{\boldsymbol{\omega}} & \boldsymbol{w}_{\boldsymbol{a}} & \boldsymbol{w}_{\boldsymbol{b}_{\boldsymbol{\omega}}} & \boldsymbol{w}_{\boldsymbol{b}_{\boldsymbol{\alpha}}} \end{bmatrix} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_t)$ is the input−process noise that follows a normal zero mean distribution with covariance $\boldsymbol{Q}_t$, and $[\times]$ is the wedge operation.

Subsequently, denoting the error state vector as:

$$\boldsymbol{\delta x}_t = \begin{bmatrix} {}^b\boldsymbol{\delta v}_b & \boldsymbol{\chi} & {}^w\boldsymbol{\delta p}_b & \boldsymbol{\delta b}_{\boldsymbol{\omega}} & \delta \boldsymbol{b}_{\boldsymbol{a}} \end{bmatrix}^\top \tag{4.2}$$

the error-state dynamics assume the following linear form:

$$\boldsymbol{\delta \dot{x}}_t = \boldsymbol{F}_t \boldsymbol{\delta x}_t + \boldsymbol{L}_t \boldsymbol{w}_t \tag{4.3}$$

with

$$\boldsymbol{F}_t = \begin{bmatrix} -{}^b\bar{\boldsymbol{\omega}}_{b[\times]} & \left({}^w\boldsymbol{R}_b^\top \boldsymbol{g}\right)_{[\times]} & \boldsymbol{0} & -{}^b\boldsymbol{v}_{b[x]} & -\boldsymbol{I} \\ \boldsymbol{0} & -{}^b\bar{\boldsymbol{\omega}}_{b[\times]} & \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{0} \\ {}^w\boldsymbol{R}_b & -{}^w\boldsymbol{R}_b{}^b\boldsymbol{v}_{b[x]} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{4.4}$$

$$\boldsymbol{L}_t = \begin{bmatrix} -{}^b\boldsymbol{v}_{b[x]} & -\boldsymbol{I} \\ -\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \tag{4.5}$$

To this end, the ESKF predict step is readily realized as:

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{f}_k^d(\hat{\boldsymbol{x}}_{k-1}^+, \boldsymbol{u}_k, \boldsymbol{0}) \tag{4.6}$$

$$\boldsymbol{P}_k^- = \boldsymbol{F}_k^d \boldsymbol{P}_{k-1}^+ \boldsymbol{F}_k^{d\top} + \boldsymbol{L}_k^d \boldsymbol{Q}_k^d \boldsymbol{L}_k^{d\top} \tag{4.7}$$

where the superscript d indicates the discretized variables at the discrete-time k, which

are obtained by means of the Euler method for simplicity. Moreover, $\hat{\boldsymbol{x}}_k^-$, $\boldsymbol{P}_k^-$ denote the ESKF mean estimate and error covariance respectively, prior to update, while $\hat{\boldsymbol{x}}_{k-1}^+$, $\boldsymbol{P}_{k-1}^+$ are the same quantities after the update at discrete-time $k-1$.

4.2.2 Measurement Model

The output model of [53], was formulated with the base velocity using the robot's kinematics and the base position, orientation obtained with a Gaussian particle filter on LIDAR measurements, all expressed in the world frame. In this chapter, besides the kinematically computed base velocity, we consider external measurements of the base position and orientation from either LIDAR Odometry (LO) or Visual Odometry (VO). Nevertheless, such measurements can potentially suffer from outliers in human daily environments due to the static-world assumption, as presented in Sec. 4.1. Thus, we distinguish the latter with the superscript o for possible outliers as:

$$\boldsymbol{y}_k^o = \overbrace{\begin{bmatrix} {}^w\boldsymbol{p}_b + \boldsymbol{n}_{\boldsymbol{p}_b} \\ {}^w\boldsymbol{R}_b \boldsymbol{e}^{\boldsymbol{n}_{b_{[\times]}}} \end{bmatrix}}^{\boldsymbol{h}^o(\boldsymbol{x}_k) \boxplus \boldsymbol{n}_k} \tag{4.8}$$

where $\boldsymbol{n}_k = \begin{bmatrix} \boldsymbol{n}_{\boldsymbol{p}_b} & \boldsymbol{n}_b \end{bmatrix}$ denote the external position and orientation measurement noise that follows normal zero mean distribution with covariance $\boldsymbol{R}_k^o$. The operator $\boxplus$ denotes the proper summation which applies to rotation matrices as composition of rotations.

On the contrary, we normally consider the kinematically computed base velocity as:

$$\boldsymbol{y}_k^n = \overbrace{{}^w\boldsymbol{R}_b {}^b\boldsymbol{v}_b + \boldsymbol{n}_{\boldsymbol{v}_b}}^{\boldsymbol{h}^n(\boldsymbol{x}_k) + \boldsymbol{n}_{\boldsymbol{v}_b}} \tag{4.9}$$

with $\boldsymbol{n}_{\boldsymbol{v}_b} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k^n)$ be the normal zero mean kinematic velocity noise with covariance $\boldsymbol{R}_k^n$. The above measurements do not accumulate leg odometry drift during the gait and are commonly not contaminated with outliers when accurate contact states are estimated [75, 76]. Thus, we distinguish them with the superscript n for nominal measurements that will be not examined for outliers.

To derive the linearization of Eq. (4.8) we consider the error exponential coordinates related with the external rotation [49], then:

$$\boldsymbol{\delta y}_k^o = \boldsymbol{H}_k^o \boldsymbol{\delta x}_k + \boldsymbol{n}_k \tag{4.10}$$

with

$$\boldsymbol{H}_k^o = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{4.11}$$

On the other hand, the linearization of (4.9) is straightforward to compute:

$$\boldsymbol{\delta y}_k^n = \boldsymbol{H}_k^n \boldsymbol{\delta x}_k + \boldsymbol{n}_{\boldsymbol{v}_b} \tag{4.12}$$

where

$$\boldsymbol{H}_k^n = \begin{bmatrix} {}^w\boldsymbol{R}_b & -{}^w\boldsymbol{R}_b \boldsymbol{v}_{b[x]} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{4.13}$$

Subsequently, the ESKF update step is formulated as:

$$\boldsymbol{\delta x}_k = \boldsymbol{K}_k^* \left(\boldsymbol{y}_k^* \boxminus \boldsymbol{h}^*(\hat{\boldsymbol{x}}_k^-) \right) \tag{4.14}$$

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- \boxplus \boldsymbol{\delta x}_k \tag{4.15}$$

$$\boldsymbol{P}_k^+ = \boldsymbol{P}_k^- - \boldsymbol{K}_k^* (\boldsymbol{H}_k^* \boldsymbol{P}_k^- \boldsymbol{H}_k^{*\top} + \boldsymbol{R}_k^*) \boldsymbol{K}_k^{*\top} \tag{4.16}$$

$$\boldsymbol{K}_k^* = \boldsymbol{P}_k^- \boldsymbol{H}_k^{*\top} (\boldsymbol{H}_k^* \boldsymbol{P}_k^- \boldsymbol{H}_k^{*\top} + \boldsymbol{R}_k^*)^{-1} \tag{4.17}$$

where the superscript * can be either o or n depending on the set of measurements considered.

4.2.3 Outlier Detection

In this section, the main result of this chapter is presented. The outlier detection framework presented in [96] is integrated with the ESKF to introduce a base estimator robust to outliers.

In order to detect outlier measurements, Wang et al. [96], utilized a beta-Bernouli distribution to probabilistically quantify whether a measurement is outlier or not. Beta-Bernouli distributions have been proved effective in various outlier resilient algorithms in the past [98, 99]. To this end, in [96] a binary indicator variable z_k was introduced. Accordingly, when z_k is one, y_k^o is a nomimal measurement while when z_k is zero, y_k^o is an outlier. The latter can be formulated as:

$$p(\boldsymbol{y}_k^o | \boldsymbol{x}_k, z_k) = \mathcal{N}(\boldsymbol{h}^o(\boldsymbol{x}_k), \boldsymbol{R}_k)^{z_k} \tag{4.18}$$

Evidently, when $z_k = 0$, Eq. (4.18) becomes a constant and cannot contribute to the state estimation, since the distribution is measurement independent.

Subsequently, in order to properly infer the indicator variable, a beta-Bernoulli hierar-

chical prior [100] is enforced. In such a way, z_k is a Bernoulli variable influenced by π_k:

$$p(z_k|\pi_k) = \pi_k^{z_k}(1-\pi_k)^{(1-z_k)} \tag{4.19}$$

where π_k follows a beta distribution:

$$p(\pi_k) = \frac{\pi_k^{e_0-1}(1-\pi_k)^{f_0-1}}{\mathcal{B}(e_0, f_0)} \tag{4.20}$$

with $\mathcal{B}$ denoting the beta function, parametrized by e_0 and f_0.

Since, z_k is modeled as a beta-Bernoulli variable, when the mean $\langle z_k \rangle$ is close to zero, e.g. 10^{-5}, we treat the measurement as an outlier and ignore it, thus:

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- \tag{4.21}$$

$$\boldsymbol{P}_k^+ = \boldsymbol{P}_k^- \tag{4.22}$$

otherwise, we weight the measurement noise $\boldsymbol{R}_k^o$ as:

$$\boldsymbol{R}_k^o = \boldsymbol{R}_k^o / \langle z_k \rangle \tag{4.23}$$

and perform the regular update as in (4.14)-(4.17).

The expectation of z_k is computed in each iteration as follows:

$$\langle z_k \rangle = \frac{p(z_k = 1)}{p(z_k = 1) + p(z_k = 0)} \tag{4.24}$$

with

$$p(z_k = 1) = c e^{\Psi(e_k) - \Psi(e_k + f_k) - \frac{1}{2} tr\left(\boldsymbol{B}_k \boldsymbol{R}_k^{o\,-1}\right)} \tag{4.25}$$

$$p(z_k = 0) = c e^{\Psi(f_k) - \Psi(e_k + f_k)} \tag{4.26}$$

where c is the normalization constant to guarantee that (4.25), (4.26) are proper probabilities, Ψ denotes the digamma function [100], and $\boldsymbol{B}_k$ is given by:

$$\boldsymbol{B}_k = \int (\boldsymbol{y}_k^o - \boldsymbol{h}^o(\hat{\boldsymbol{x}}_k^+))(\boldsymbol{y}_k^o - \boldsymbol{h}^o(\hat{\boldsymbol{x}}_k^+))^\top p(\hat{\boldsymbol{x}}_k^+) d\boldsymbol{x}_k \tag{4.27}$$

The integral in (4.27) is not straightforward to compute in the general nonlinear Gaussian case. In [96] the cubature rules [101] to obtain an approximate solution are used. In

the context of the EKF, (4.27) can be derived analytically as:

$$\begin{aligned}\boldsymbol{B}_k = \boldsymbol{y}_k^o \boldsymbol{y}_k^{o\top} - 2\boldsymbol{y}_k^o \boldsymbol{h}^o(\hat{\boldsymbol{x}}_k^+)^\top \\ + \boldsymbol{h}^o(\hat{\boldsymbol{x}}_k^+)\boldsymbol{h}^o(\hat{\boldsymbol{x}}_k^+)^\top + \boldsymbol{H}_k^o \boldsymbol{P}_k^+ \boldsymbol{H}_k^{o\top}\end{aligned} \tag{4.28}$$

The proof is given for completeness in the section 9.6 of appendix B.

Finally, e_k, f_k are updated in each iteration as:

$$e_t = e_0 + \langle z_k \rangle \tag{4.29}$$

$$f_t = f_0 + 1 - \langle z_k \rangle \tag{4.30}$$

The proposed robust Gaussian ESKF (RGESKF) is summarized in Algorithm 1. We note, that no further knowledge of the measurement distribution [95] other than the covariance $\boldsymbol{R}_k^o$ is needed or empirically obtained thresholds as in [56] are required to perform outlier detection. The only tunable parameters are the beta-Bernoulli prior parameters e_0 and f_0. Experimentally, e_0 and f_0 have been set to 0.9 and 0.1 respectively. The latter values have been used in all conducted experiments, including real tests with the two robots (cf. Sec. 4.3 below) and in our open-source implementation [90]. As also stated in [96], we observed that the outlier detection process is insensitive to the latter parameters as long as $e_0/(e_0+f_0)$ is close to 1. Presumably, this is the case, since it is more probable to observe a nomimal measurement rather than an outlier.

Algorithm 1: Robust Gaussian ESKF

Data: $\boldsymbol{y}^o_{1:T}, \boldsymbol{y}^n_{1:T}, \hat{\boldsymbol{x}}_0, \boldsymbol{P}_0, \boldsymbol{Q}_{1:T}, \boldsymbol{R}^o_{1:T}, \boldsymbol{R}^n_{1:T}$
Result: $\hat{\boldsymbol{x}}^+_k,\ \boldsymbol{P}^+_k$ for $k = 1:T$

```
for k = 1, ..., T do
    Compute x̂_k^- and P_k^- via (4.6), (4.7);
    Initialize i = 0, e_0 = 0.9, f_0 = 0.1, and ^(i)z_k=1;
    repeat
        Update R_k^o with (4.23);
        i = i + 1;
        if ^(i-1)z_k > 10^-5 then
            Update ^(i)x̂_k and ^(i)P_k^+ via (4.14)-(4.17) with position and orientation;
            Update ^(i)z_k via (4.24);
            Update ^(i)e_k and ^(i)f_k via (4.29), (4.30);
        else
            Update ^(i)x̂_k and ^(i)P_k^+ via (4.21), (4.22);
            break;
        end
    until ||^(i)x̂_k ⊟ ^(i-1)x̂_k|| < 10^-3;
    x̂_k^+ = ^(i)x̂_k^+ and P_k^+ = ^(i)P_k^+;
    Update x̂_k and P_k^+ via (4.14)-(4.17) with velocity;
end
```

Figure 4.2: Illustration of conducted experiments – **Left**: WALK-MAN v2.0 and VO outliers, **Middle**: NAO and VO outliers, **Right**: NAO and LO outliers.

4.3 Results

In this section, we outline representative results that demonstrate the accuracy and efficiency of the proposed scheme under real world conditions. Two actual humanoids were employed in our experiments, the full-size 29DoF WALK-MAN v2.0 humanoid [97] and a mini-size NAO robot. The WALK-MAN v2.0 robot uses the walking module introduced in [102, 103], with step time of $0.8s$, using the XBotCore [104] and OpenSoT [105] control infrastructure in a $500Hz$ control-loop. The walking module utilized with the NAO robot is based on [67] with a step time of $0.4s$ and achieves a control-loop of $100Hz$. The IMUs noise standard deviations used in our experiments are shown in Table 4.1. For the WALK-MAN v2.0 VectorNav VN-100 IMU, we employed the noise densities given by the manufacturer at $200Hz$, while for the IMU utilized in the NAO experiments an Allan variance analysis [106] was performed with 13 hour stationary data at $100Hz$ to properly derive the values.

Additionally, joint angle measurements are available at $200Hz$ for the WALK-MAN v2.0 robot and at $100Hz$ for NAO.

Table 4.1: IMU Noise stds

	$\boldsymbol{w_\omega}(\frac{rad}{s})$	$\boldsymbol{w_\alpha}(\frac{m}{s^2})$	$\boldsymbol{w_{b\omega}}(\frac{rad}{s^2})$	$\boldsymbol{w_{b\alpha}}(\frac{m}{s^3})$
WALK-MAN	9.77e-4	2.21e-2	1.53e-5	2.43e-4
NAO	5.63e-3	1.58e-2	9.66e-4	4.33e-3

To facilitate a quantitative assessment, we compare the proposed RGESKF, to the ESKF without outlier detection, and to an ESKF where the outlier detection method in [74] is employed. The latter is termed as ESKF-TH since in each experiment we had to fine tune in advance a probabilistic threshold TH to achieve accurate detection according to the Mahalanobis distance d_M:

$$d_M = \left(\boldsymbol{y}_k^o - \boldsymbol{h}^o(\hat{\boldsymbol{x}}_k^-)\right)^\top \boldsymbol{S}_k^{-1} \left(\boldsymbol{y}_k^o - \boldsymbol{h}^o(\hat{\boldsymbol{x}}_k^-)\right) \tag{4.31}$$

with

$$\boldsymbol{S}_k = \boldsymbol{H}_k^o \boldsymbol{P}_k^+ \boldsymbol{H}_k^{o\top} + \boldsymbol{R}_k^o \tag{4.32}$$

In all our experiments, the aforementioned estimation schemes achieved real-time execution at the corresponding IMU rates.

4.3.1 VO Outliers

In the case of VO, we conducted two independent experiments, one with the WALK-MAN v2.0 while the ground-truth was recorded with an OptiTrack motion capture system and another one with a NAO robot navigating to a desired position in space. In the former case the PointGrey BlackFly BFLY-U3-23S6C camera at 40fps and 1080p resolution was used while in the latter, the Matrix-Vision mvBlueFOX-MLC-200w running at 30fps and VGA was utilized. Both cameras are monocular and global-shutter. To obtain the VO in both cases we used SVO [91]. The measurement noise standard deviations used in our VO experiments, are listed in Table 4.2.

In the WALK-MAN v2.0 experiment, shown in Fig. 4.2 (left), a human unexpectedly crosses the Field of View (FoV) of the robot, causing sudden changes in the image intensity levels, and furthermore removes an object that is a strong feature source for VO. As a result, the scene changes rather drastically and henceforth, VO diverges while generating consecutive outlier measurements.

In Figure 4.3, the estimated 3D position and orientation errors w.r.t ground-truth for all three employed schemes is illustrated. Notice, at $t = 17.6s$ where the human appears in the FoV, VO starts to misbehave and at $t = 19.9$ when static scene changes, VO eventually

Table 4.2: VO/LO and Kinematic Measurement Noise stds

	$\boldsymbol{n}_{\boldsymbol{p}_b}(m)$	$\boldsymbol{n}_b(rad)$	$\boldsymbol{n}_{\boldsymbol{v}_b}(\frac{m}{s})$
WALK-MAN - VO	0.035	0.05	0.015
NAO - VO	0.04	0.05	0.013
NAO - LO	0.04	0.05	0.013

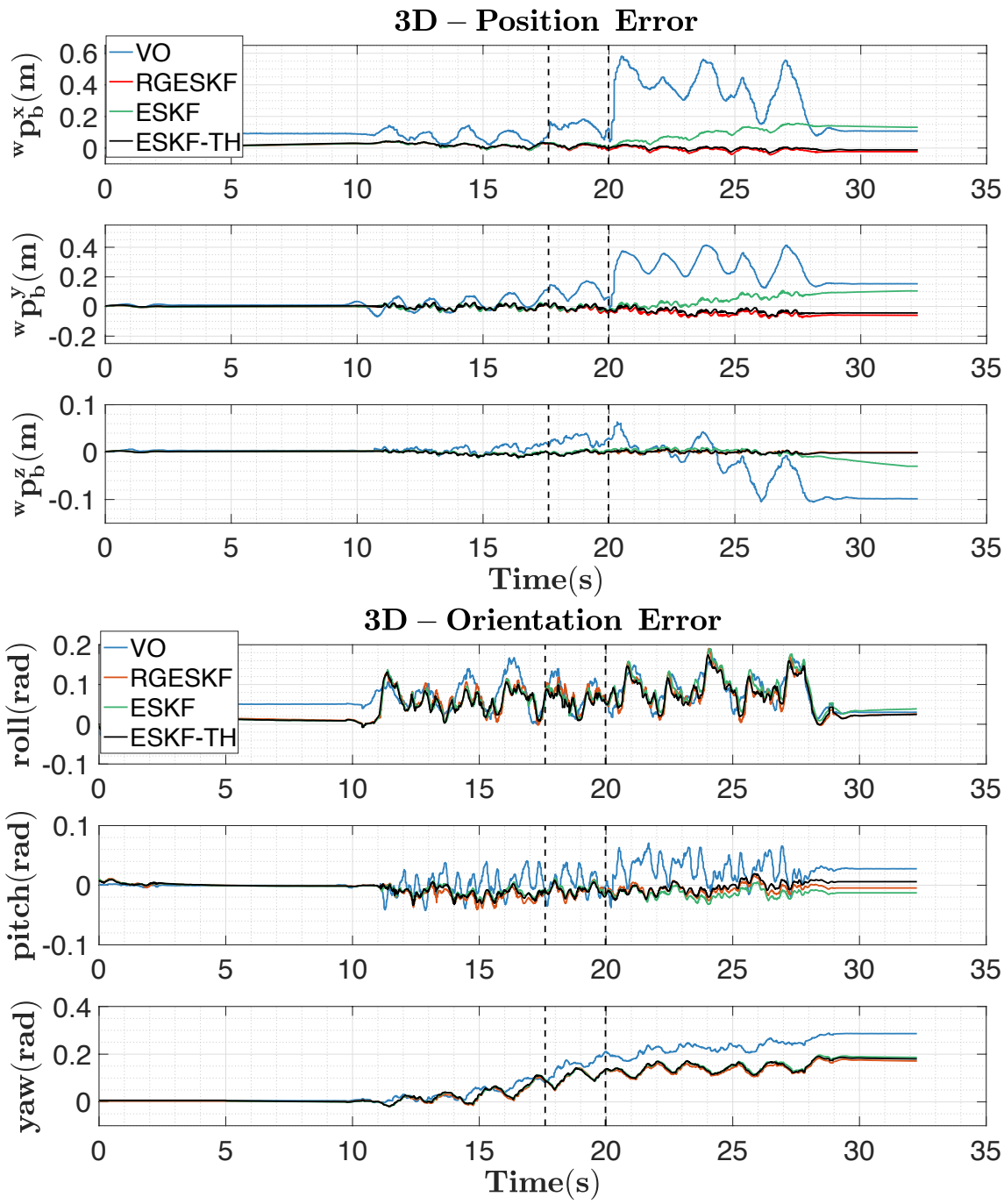

Figure 4.3: **Top**: 3D - position error, **Bottom**: 3D - orientation error for the WALKMAN 2.0 - VO experiment, blue lines indicate the VO, red lines the proposed Robust Gaussian ESKF, green lines the ESKF, and black lines the finely tuned ESKF-TH. The first vertical dotted line corresponds to when a human enters the FoV of the camera, while the second one when he removes a strong feature source for VO which diverges.

Table 4.3: RMSE for the WALK-MAN v2.0 VO experiment

	${}^{w}\boldsymbol{p}_b^x$	${}^{w}\boldsymbol{p}_b^y$	${}^{w}\boldsymbol{p}_b^z$	roll	pitch	yaw
VO	0.217	0.162	0.043	0.075	0.022	0.160
ESKF	0.071	0.045	0.009	0.066	0.011	0.105
RGESKF	0.018	0.033	0.003	0.062	0.011	0.098
ESKF-TH	0.017	0.025	0.003	0.059	0.010	0.102

diverges. Consequently, the ESKF without outlier detection diverges as well and thus large positional errors are recorded. Nevertheless, this is not the case for the RGESKF and the ESKF-TH, where low errors were observed for both the position and the orientation. To finely tune the ESKF-TH, we had to run the filter and log the Mahalanobis distances (4.31), in order to determine a proper threshold TH. A value of $\mathrm{TH} = 23$ achieved the lowest estimation error. On the contrary our approach, RGESKF, which does not rely on finely tuned threshold or prior knowledge on the measurements, achieved very accurate and similar results to the tuned in advanced ESKF-TH. The Root-Mean Square Error (RMSE) for this experiment is shown in Table 4.3. All employed schemes realized errors in the case of yaw estimation since it is unobservable [56, 57].

Similarly in the NAO's case, a human, present in the scene removes a strong feature source for VO, while the robot is walking. For clarity, this is illustrated in Fig. 4.2 (middle). Since, we do not have ground-truth data available for the NAO experiments and given that a fine tuned ESKF-TH can yield pretty accurate estimation, as evident by our previous experiment, we assume it as baseline to compare to. Subsequently, to properly derive the threshold needed, we computed (4.31) at $t = 39.6s$, the exact time when the human changes the scenery. The previous, was found to be $\mathrm{TH} = 16$. Figure 4.4 demonstrates the 3D position and orientation error w.r.t the ESKF-TH. As illustrated, the ESKF realizes large errors for both positional and rotational quantities, when VO diverges at $t = 39.6s$. However, once again the RGESKF yielded practically identical results to the ESKF-TH as also evident in Table 4.4, where the RMSE is presented.

Table 4.4: RMSE w.r.t ESKF-TH for the NAO VO experiment

	${}^{w}\boldsymbol{p}_b^x$	${}^{w}\boldsymbol{p}_b^y$	${}^{w}\boldsymbol{p}_b^z$	roll	pitch	yaw
VO	0.246	0.653	0.227	0.066	0.246	0.152
ESKF	0.238	0.643	0.223	0.037	0.061	0.146
RGESKF	2.8e-6	2.6e-7	2.3e-6	6.8e-7	1.6e-6	9.2e-6

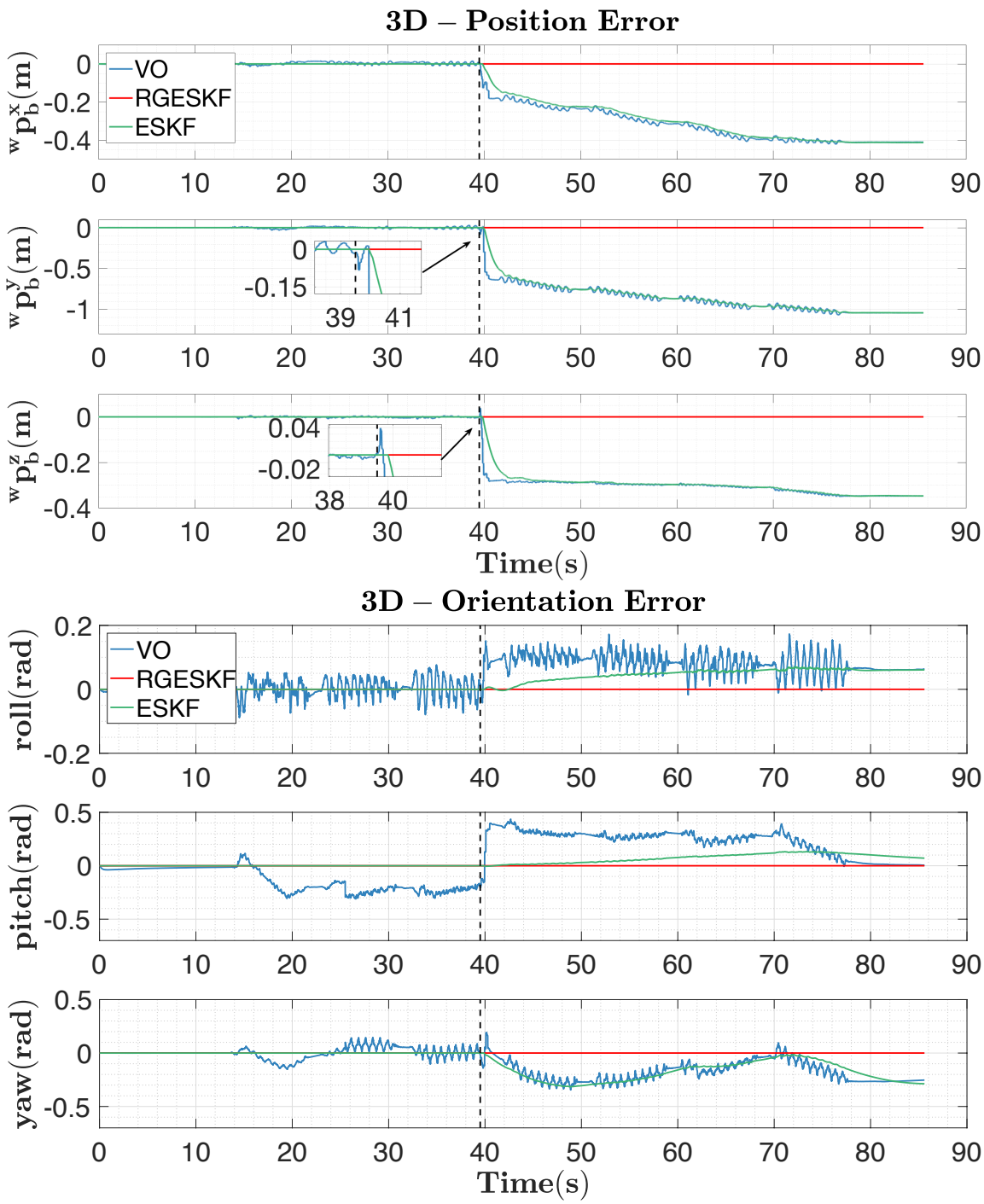

Figure 4.4: **Top**: 3D - position error, **Bottom**: 3D - orientation error for the NAO - VO experiment, blue lines indicate the VO, red lines the proposed Robust Gaussian ESKF, and green lines the ESKF. The vertical dotted line specifies when a human removes a strong feature source for VO which diverges.

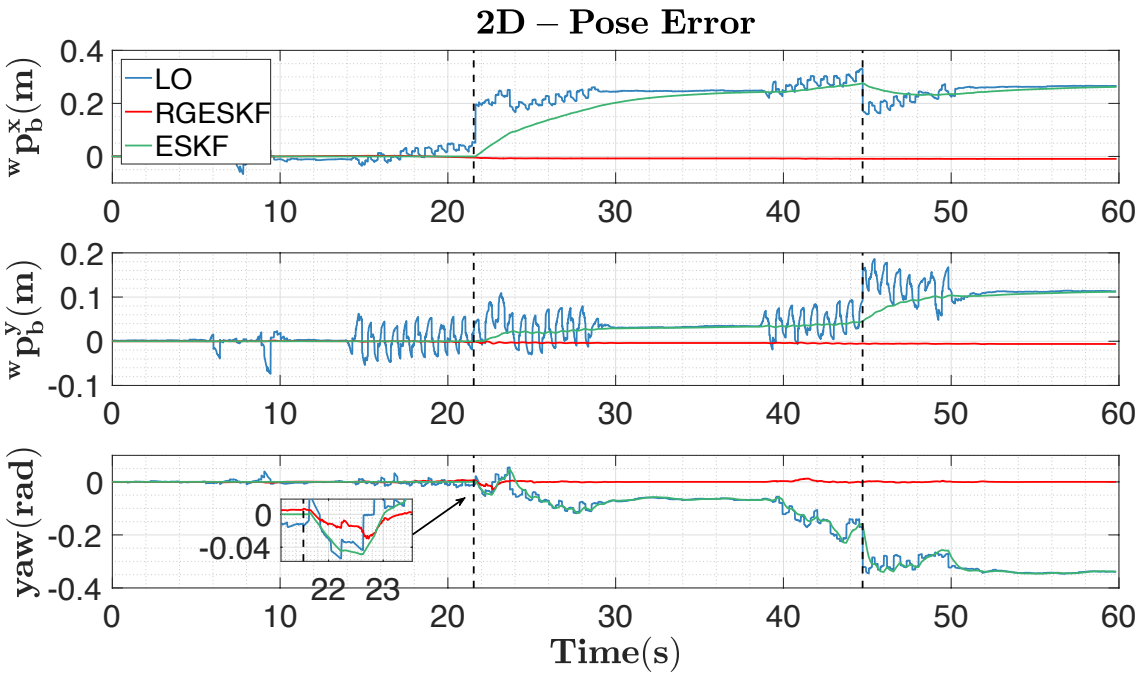

Figure 4.5: 2D - pose error, for the NAO - LO experiment, blue lines indicate the LO, red lines the proposed Robust Gaussian ESKF, and green lines the ESKF. The vertical dotted lines corresponds to when a human suddenly covers the robot's LiDAR with a box, causing LO to diverge.

4.3.2 LO Outliers

Next, we examine how LO outlier measurements can degrade the estimation performance. To do so, we utilize an RP-LiDAR360 mounted on NAO's head to obtain planar scans every $5Hz$. Subsequently, we employed RF2O [93] to compute the 2D pose e.g. position x,y and yaw, with scan matching. The measurement noise assumed in this experiment is shown at Table 4.2.

In order to generate LO outliers, a human covers the spinning laser while NAO walks, as depicted in Fig. 4.2 (right). This corresponds to the scenario where a robot in motion is suddenly surrounded by people. To this end, the static world used to derive the LO is drastically changed which in turn gives rise to outliers.

In the conducted experiment, NAO is commanded to walk straight, stop, and then walk straight again. While walking a human covers the LiDAR twice to generate LO outliers. As previously, we compare the estimation results to the ESKF-TH, which we have accurately tuned in advance as before. A threshold of $\mathrm{TH} = 9$ was experimentally found to be sufficient for this specific experiment. The 2D pose error is shown in Figure 4.5. Time $t = 21.5s$ marks the instant where the human covers the LiDAR for the first time. At that time, a large

Table 4.5: RMSE w.r.t ESKF-TH for the NAO LO experiment

	${}^{w}\boldsymbol{p}_b^x$	${}^{w}\boldsymbol{p}_b^y$	yaw
LO	0.197	0.066	0.174
ESKF	0.180	0.054	0.173
RGESKF	0.004	0.002	0.006

jump in the LO position in x axis is recorded which in turn causes large error to the ESKF estimation in the same axis. Subsequently, after $23.5s$ the human covers the LiDAR one more time. This time larger errors are evident in the base's y position and the base's yaw angle causing again the ESKF to misbehave. Interestingly, the proposed scheme was proven to be robust and automatically ignore the inaccurate LO measurements. The RMSE for this particular experiment is indicated in Table 4.5. As demonstrated, the RGESKF yields a similar estimation result for all quantities of interest when compared to a finely tuned ESKF-TH. All our experiments are illustrated in high quality at `https://youtu.be/ojogeY3xSsw`.

4.3.3 Qualitative Assessment

As evident, the proposed RGESKF is characterized by high accuracy and strong outlier rejection capabilities. The latter hold true, even when consecutive VO/LO outlier measurements were observed. Additionally, no prior knowledge of the measurement distributions [95] or finely tuned thresholds [74] are required for the success of the proposed scheme. On the contrary, notice that the ESKF-TH needed three different thresholds, one for each experiment, to achieve accurate performance. This is evident by (4.31), where an optimal threshold depends on the measurement noise $\boldsymbol{R}_k^o$ and the error-state uncertainty $\boldsymbol{P}_k^+$. In addition, in all our VO/LO experiments, the outlier detection part in Algorithm 1 took at most three iterations to complete. Thus, in the open-source released implementation [90], we loop three times instead of computing in every iteration the condition in line 15. Moreover, it is noteworthy that in the VO experiments the initial derived SVO orientation can be erroneous. We suspect this is probably due to a) inaccurate scale initialization, b) imperfect extrinsic and intrinsic calibrations. Nevertheless, as also seen in the results, this does not degrade the estimation accuracy since the fused IMU and kinematically computed base velocity measurements also carry information that contribute to the orientation estimation. Furthermore, it is important to clarify that the RGESKF does not only detect and reject outliers as the ESKF-TH does, but automatically weights the measurement noise according to (4.23) in order to avoid information loss when non-ideal/non-outlier measurements arrive. Finally, the proposed method can be appropriately employed to other robotic platforms, such as Unmanned Aerial Vehicles (UAVs), which also utilize the ESKF [107] for state estimation.

4.4 Conclusion

Prominent examples of VO/LO approaches readily assume that the world in which the robot acts, is static. Nevertheless, to enable humanoids co-exist with humans in dynamically changing environments, their state-estimation schemes must be robustified. In this chapter, we tackled the presence of VO/LO outlier measurements in base estimation by proposing the RGESKF. After mathematically establishing the proposed scheme, we provided a quantitative and qualitative assessment with two robots, namely a full-size WALKMAN v2.0 humanoid and a mini-size NAO robot, demonstrating the accuracy and efficiency of the proposed scheme in real-world conditions. Finally, in order to reinforce further research endeavours, we released our implementation as an open-source ROS/C++ package [90].

However, when computing the base velocity needed in the EKF measurement model, we've implicitly assumed that the contact status of the legs is known in advanced. Contact detection is a topic of vital importance in legged robotics with application in gait control and state estimation. In the next chapter, we will propose an unsupervised learning framework that accurately estimates the gait phase and thus, implicitly the contact status.

Chapter 5
Unsupervised Gait-Phase Estimation

Look up at the stars and not down at your feet. Try to make sense of what you see, and wonder about what makes the universe exist. Be curious.

Stephen Hawking (1942–2018)

Contact detection is an important topic in contemporary humanoid robotic research. Up to date control and state estimation schemes readily assume that feet contact status is known in advance. In this chapter, we elaborate on a broader question: *in which gait phase is the robot currently in?* We introduce an unsupervised learning framework for gait phase estimation based solely on proprioceptive sensing, namely joint encoder, inertial measurement unit and force/torque data. Initially, a meaningful physical explanation on data acquisition is presented. Subsequently, dimensionality reduction is performed to obtain a compact low-dimensional feature representation followed by clustering into three groups, one for each gait phase. The proposed framework is qualitatively and quantitatively assessed in simulation with ground-truth data of uneven/rough terrain walking gaits and insights about the latent gait phase dynamics are drawn. Additionally, its efficacy and robustness is demonstrated when incorporated in leg odometry computation. Since our implementation is based on sensing that is commonly available on humanoids today, we release an open-source ROS/Python package to reinforce further research endeavors.

5.1 Aim and Contribution

Contemporary research approaches in motion planning, control and estimation for legged robots readily assume that contact states are known a priori. Whole body control [61–63] and gait planning [36, 64, 65] are explicitely based on contact models. Even when simplified dynamical models are employed in the design [10,66,67], the contact state is implicitly considered in the computation of the Center of Pressure (COP). Nevertheless, detecting ground contact and furthermore, determining which is the support leg e.g. experiences a

rigid contact with the ground, is non-trivial in legged robotics [68]. In all state-of-the-art approaches, the objective was to determine whether a specific foot is in contact or not. In this chapter we raise a broader question: *in which gait phase is the robot currently in?* To this end, we propose a holistic framework based on unsupervised learning from proprioceptive sensing that accurately and efficiently addresses this problem. The contributions of this work to the state of art are:

- Robust classification to three gait phases, namely Left Single Support (LSS), Double Support (DS), and Right Single Support (RSS).
- Fusion of joint encoder, IMU, and F/T measurements in a solely unsupervised learning framework.
- A meaningful physical explanation on this particular data selection is provided.
- It is demonstrated that gait phase dynamics are low-dimensional which is another indication pointing towards locomotion being a low dimensional skill.
- Given that the proposed framework utilizes measurements from sensors that are commonly available on humanoids nowadays, we offer the Gait-phase Estimation Module (GEM), an open-source implementation to reinforce further research endeavors [108].

5.2 Unsupervised Gait Phase Estimation

In this section the proposed gait phase estimation framework is presented. The latter illustrated in Figure 5.1, is completely unsupervised, meaning no labels or ground-truth data are required in training. Initially, we argue on the suitability of unsupervised learning for the task at hand. Next, a meaningful physical explanation on data acquisition is given. Subsequently, dimensionality reduction is performed to obtain a compact low-dimensional feature representation followed by clustering into three groups, one of each gait phase.

5.2.1 Why Unsupervised Learning?

Let the Newton-Euler equations of motion for a humanoid robot:

$$m(\ddot{\boldsymbol{c}} - \boldsymbol{g}) = \sum_i \boldsymbol{f}_i \tag{5.1}$$

$$m\boldsymbol{c} \times (\ddot{\boldsymbol{c}} - \boldsymbol{g}) + \dot{\boldsymbol{L}} = \sum_i \boldsymbol{s}_i \times \boldsymbol{f}_i + \boldsymbol{\tau}_i \tag{5.2}$$

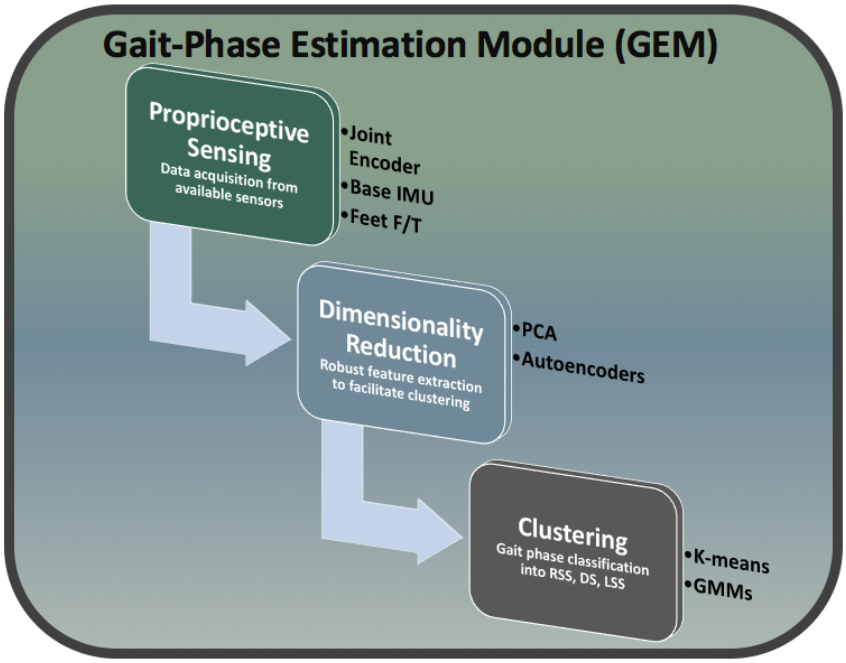

Figure 5.1: Gait-Phase Estimation Module.

where $\boldsymbol{c}$ is the CoM position, $\ddot{\boldsymbol{c}}$ is the CoM acceleration, $\dot{\boldsymbol{L}}$ is the rate of angular momentum, $\boldsymbol{f}_i$ and $\boldsymbol{\tau}_i$ are the contact forces/moments, $\boldsymbol{s}_i$ are the contact points, m is the mass of the robot, and $\boldsymbol{g}$ is the gravity vector.

In addition, for a foot to experience a rigid contact with the ground and do not slip or rotate, the following inequalities must hold:

$$\sqrt{(f^x)^2 + (f^y)^2} \leq \mu_{x,y} f^z \tag{5.3}$$

$$-\tau^y / f^z \leq p^x \tag{5.4}$$

$$\tau^x / f^z \leq p^y \tag{5.5}$$

$$|\tau^z| \leq \mu_z f^z \tag{5.6}$$

where $\mu_{x,y}, \mu_z$ are the contact friction coefficients, and $\boldsymbol{p}$ is the COP.

Accordingly, while walking, (5.1) and (5.2) with the contact constraints (5.3)-(5.6) describe the humanoid's motion during both SS and DS phases. Nevertheless, the gait phase cannot be derived analytically from the above equations since the contact friction coefficients are commonly unknown and depend on environmental properties. To this end, the gait phase needs to be inferred from data. In addition, since ground-truth contact labels are very hard to obtain even in simulation, unsupervised learning is appropriate for the task at hand.

5.2.2 Data Acquisition

As evident by (5.1) and (5.2), the current contact status, and hence the gait phase, gives rise to the CoM dynamics and the angular momentum around the CoM. Furthermore, during various omni-directional gaits the CoM motion is similar to the body motion [87], since most of the humanoid's mass commonly exists above the legs. Therefore, the angular momentum around the CoM can be readily approximated by the body's angular momentum [49].

Thus, it is meaningful to infer the gait phase based on a) the kinematically computed CoM position, relative to the base frame, ${}^{b}\boldsymbol{c}$, b) the body's θ^{roll} and θ^{pitch} angles, as estimated with an IMU, and c) the contact wrenches $({}^{\text{L,R}}\boldsymbol{f}, {}^{\text{L,R}}\boldsymbol{\tau})$ as measured by F/T in the left and right leg, respectively. All aforementioned quantities can be readily computed or directly measured from three different sensor sources, namely joint encoders, IMU and F/T.

During omni-directional walking, ${}^{b}\boldsymbol{c}$, θ^{roll} and θ^{pitch} realize oscillatory behaviors. In addition, when taking the differences $\Delta\boldsymbol{f} = {}^{\text{L}}\boldsymbol{f} - {}^{\text{R}}\boldsymbol{f}$ and $\Delta\boldsymbol{\tau} = {}^{\text{L}}\boldsymbol{\tau} - {}^{\text{R}}\boldsymbol{\tau}$ similar oscillatory patterns arise. This is illustrated in Figure 5.2, where four indicative quantities are shown for the sake of brevity. Therefore, the latter quantities are excellent candidates to facilitate clustering [109]. Note, that we don't include the θ^{yaw} angle in our data, since it is not observable [56, 57] and does not provide useful information about the gait phase.

Subsequently, to remove possible offsets in our data caused either by kinematic or IMU biases and to consider a more dynamical aspect of the data, we take the sequential differences:

$$ {}^{b}\delta\boldsymbol{c}_n = {}^{b}\boldsymbol{c}_n - {}^{b}\boldsymbol{c}_{n-1} \tag{5.7} $$

$$ \delta\theta_n^{\text{roll}} = \theta_n^{\text{roll}} - \theta_{n-1}^{\text{roll}} \tag{5.8} $$

$$ \delta\theta_n^{\text{pitch}} = \theta_n^{\text{pitch}} - \theta_{n-1}^{\text{pitch}} \tag{5.9} $$

where the subscript n stands for the n-th measurement. Summarizing the dataset $\mathcal{D}$ is 11 dimensional:

$$ \mathcal{D} = \left\{ \underbrace{{}^{b}\delta\boldsymbol{c}_n, \delta\theta_n^{\text{roll}}, \delta\theta_n^{\text{pitch}}, \Delta\boldsymbol{f}, \Delta\boldsymbol{\tau}}_{\boldsymbol{x}^{(n)}} \right\} \in \mathbb{R}^{11\times N} \tag{5.10} $$

with N be the size of the dataset and $\boldsymbol{x}^{(n)}$ be the n-th data in the set. Notice, all data are normalized in the range $[-1,\ 1]$ after acquisition.

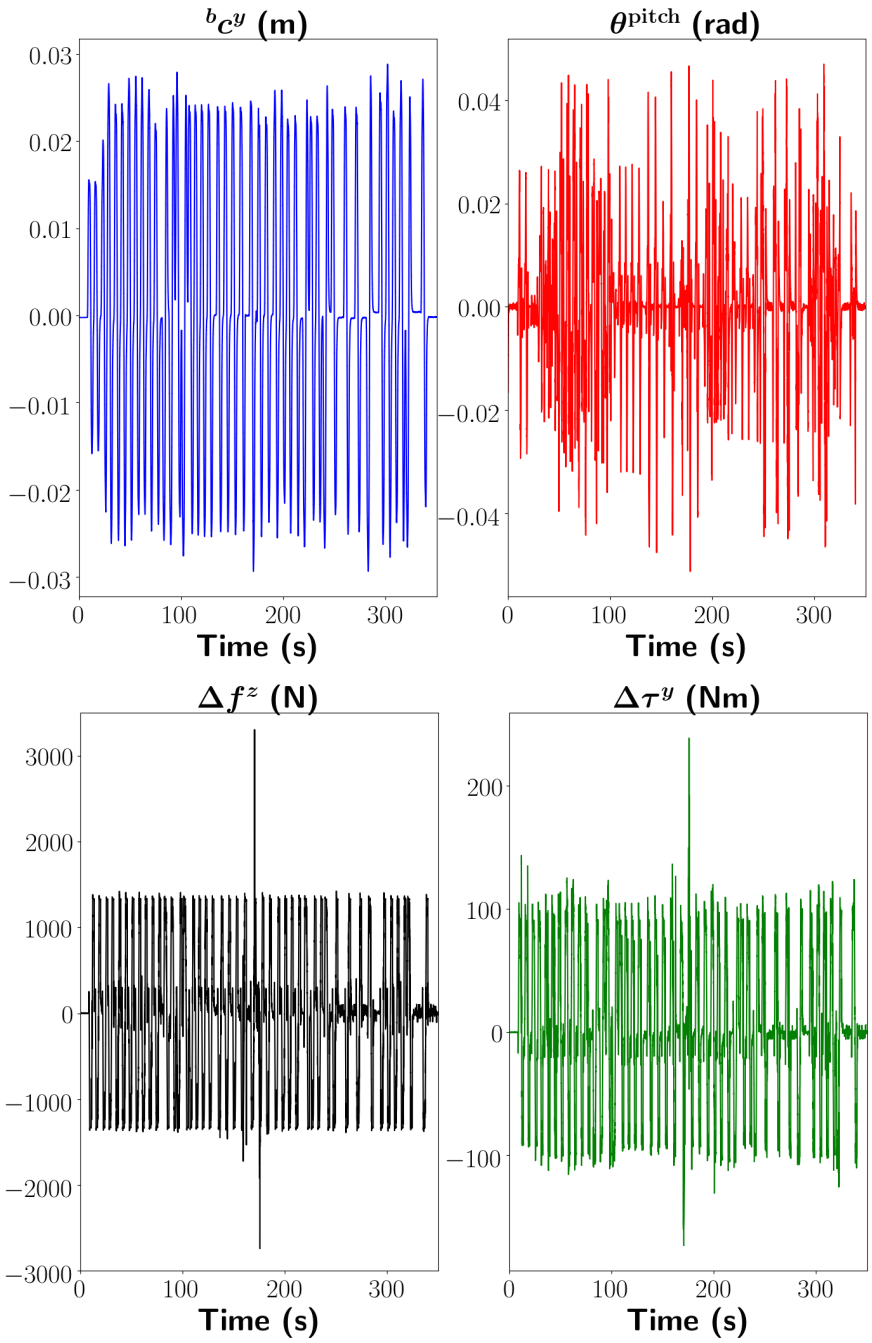

Figure 5.2: Indicative segment of the dataset $\mathcal{D}$. **Top**: lateral CoM and pitch trajectories. **Bottom**: Differential normal force and lateral torque for omni-directional gaits on uneven/rough terrain with the NASA's Valkyrie humanoid in Gazebo. Oscillatory patterns emerge in all quantities.

5.2.3 Dimensionality Reduction

In order to get a compact representation of the data, reduce the noise in the dataset, extract useful features from it and facilitate their visualization, we search for an embedded low-dimensional manifold.

Dimensionality reduction can be beneficial when a representative latent space is hidden in the data. Furthermore, as we will see in section 5.3, it is mandatory for the success of the proposed framework since noisy data/outliers tend to degrade the clustering accuracy.

In this chapter, we employ two dimensionality reduction approaches, namely the Principal Component Analysis (PCA) and the autoencoders [110]. The former is a linear projection method while the latter employs Neural Networks (NN) that can discover nonlinear relationships between the high dimensional data and the latent space, yielding potentially a more accurate low-dimensional representation. Both approaches minimize the reprojection error:

$$J = \|\mathcal{D} - \hat{\mathcal{D}}\|_F^2 \tag{5.11}$$

where F is the Frobenius norm and $\hat{\mathcal{D}}$ is the reconstructed dataset. A notable difference is that PCA computes an orthonormal mapping, thus the linear projection, while autoencoders generate a nonlinear mapping with no such constraints.

5.2.4 Clustering

Based on the obtained low-dimensional space we cluster the latent data into three groups, one for each gait phase, namely LSS, DS, RSS.

K-means, being a common clustering technique, searches for cluster centers $\boldsymbol{l}_k$ by minimizing the squared euclidean distance of each data from a cluster center:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} \|\boldsymbol{x}^{(n)} - \boldsymbol{l}_k\|^2 \tag{5.12}$$

where K is the number of clusters, in our case $K = 3$, e.g. the number of gait phases.

Consequently, K-means and its variants can only identify spherical clusters with the same radius. Thus, in order to cluster data that don't belong to spherically shaped clusters, i.e. ellipsoids, a more robust approach is needed. In addition, K-means does not take into account the density of the data, meaning that cluster centers will be always located where dense data groups are, ignoring any underlying data distribution.

To this end, we employ the Gaussian Mixture Models (GMMs) for clustering. GMMs is a kernel density estimation method which is commonly applied to approximate the probability distribution of multi-modal data as a sum of Gaussian distributions. Training is done with Expectation-Maximization (EM) in two steps:

E-step: Compute the responsibilities:

$$\gamma_k^{(n)} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}^{(n)} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}^{(n)} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{5.13}$$

with $\gamma_k^{(n)}$ be the probability of the n-th data to be generated from the k-th Gaussian distribution.

M-step: Compute the model parameters:

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} \boldsymbol{x}^{(n)} \tag{5.14}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_k)(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_k)^\top \tag{5.15}$$

$$\text{with} \;\; \pi_k = \frac{N_k}{N} \quad \text{and} \quad N_k = \sum_{n=1}^{N} \gamma_k^{(n)} \tag{5.16}$$

where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mean and covariance of the k-th Gaussian, respectively.

With GMMs the cluster shape is directly related to the estimated cluster covariance, thus flexible and elliptical clusters can be obtained. In addition, for each data item a membership probability is assigned to declare its assignment to any of the three clusters. On the contrary, in the case of K-means, each data is directly assigned to a specific cluster.

Table 5.1: Clustering Accuracy in 11-D.

	DS	LSS	RSS
K-means	66%	100%	100%
GMMs	66%	100%	100%

5.3 Results

The proposed framework has been implemented and experimentally validated. In the current section we present quantitative results that demonstrate the accuracy in uneven/rough terrain gaits. Subsequently, the efficiency is evaluated when computing leg odometry in a simulated gait. Since, to the best of our knowledge, there aren't contemporary works that directly estimate the gait phase, our comparisons are in all cases against ground-truth data.

5.3.1 Experimental Validation

Initially, to obtain quantitative assessment results, we performed omni-directional gaits on uneven/rough terrain with the NASA's Valkyrie robot in Gazebo for approximately 15 minutes to record the training dataset. The IMU, joint encoder, and F/T measurements are available at a $500Hz$ rate, thereby resulting in a dataset of approximately 450000 entries. In addition, i.i.d. Gaussian noise is added to the measurements to provide an accurate assessment with realistic noise levels.

Furthermore, we logged the ground-truth forces $\boldsymbol{f}_{gt}$, torques $\boldsymbol{\tau}_{gt}$, and COP $\boldsymbol{p}_{gt}$ that Valkyrie experienced in each leg while walking, as well as the true contact friction coefficients $(\mu_{x,y}, \mu_z)$. The latter varied from 0.7 to 1.5 depending on the environment surfaces. Subsequently, to generate the gait phase ground-truth labels, at each sample we evaluated the contact constraints (5.3)-(5.6). If those hold for both legs, then both feet experience a rigid contact with the ground and the robot is in DS. Accordingly, if the latter are true for the left or the right leg only, then the robot is in LSS or RSS, respectively. The ground-truth labels serve only to quantify the accuracy of the clustering methods.

As already detailed in section 5.2.3 the raw 11-D data tend to carry redundancies and be quite noisy, mostly due to outliers. The latter are observed in abnormal gaits, i.e. when the robot contact the ground earlier than expected very large GRFs are measured. This was also experimentally verified by the fact that clustering with the original 11-D dataset was erroneous as shown in Table 5.1, where a significant error in the DS phase is observed for both K-means and GMMs.

Accordingly, the intermediate step of dimensionality reduction plays an essential role in conditioning the data. In order to determine the appropriate latent dimension, data were projected with PCA to 3D and 2D and in both cases the reprojection error and the

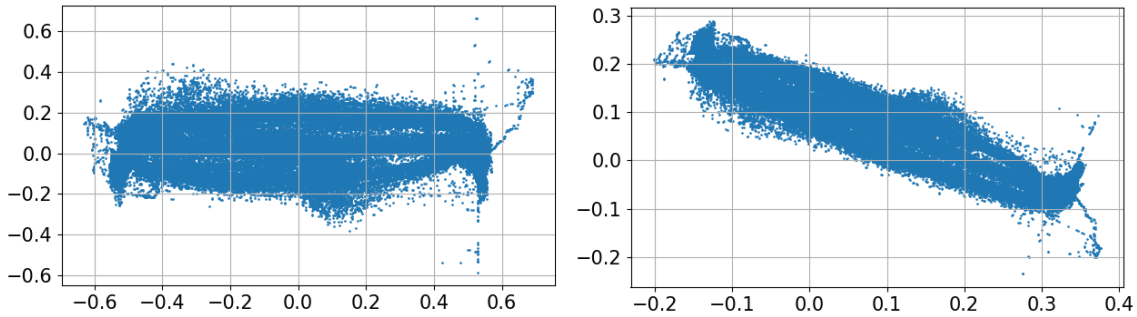

Figure 5.3: Dimensionality reduction. **Left**: Latent Space generated with PCA. **Right**: Latent Space generated with Autoencoders.

explained variance ratio of each principal component [109] were computed. Interestingly, we discovered that the 2D reprojection error was $10e-04$ and pretty similar to the one obtained with the 3D projection namely, $7.81e-04$. Furthermore, the first principal component could represent 90.6% of the data variance while the second one another 3.7%. To this end, we concluded that a 2D latent space, shown in Figure 5.3 left, can effectively represent the training dataset.

Subsequently, autoencoders were employed to investigate whether a lower reprojection error could be achieved when reducing to 2D. While experimenting with the NN structure and activation functions, we observed that good results were obtained when the encoder was formulated with two hidden layers and linear activation functions which facilitated reduction to five and two dimensions sequentially. For the decoder part the reverse structure was used. The obtained reprojection error was $8.83e-04$, which is almost identical to the one obtained with PCA. Additionally, the derived latent-space has the same shape as the one computed with the PCA transformation but different orientation and scale (Figure 5.3 right). Accordingly, it is deduced that linear projection is sufficient for this particular data selection and, therefore, is adopted for simplicity in the proposed framework. The extracted latent-space with the ground-truth labels is illustrated in Figure 5.4.

Next, the 2D-latent space is clustered into three groups with K-means as shown in Figure 5.5. When contrasting the obtained clusters to the ground-truth ones, a significant error in the DS phase can be observed. This is due to K-means clustering the space into three equal parts. Furthermore, this is also evident when computing the confusion matrix (Figure 5.6) where as expected all data belonging to either LSS or RSS have been correctly assigned but at the cost of misclassifying the DS phase.

To significantly improve the clustering accuracy, one needs to search for non-spherical clusters. Thus, the GMMs, as presented in section 5.2.4, are employed to perform probabilistic clustering. The obtained clusters are illustrated in Figure 5.7. Although, visually

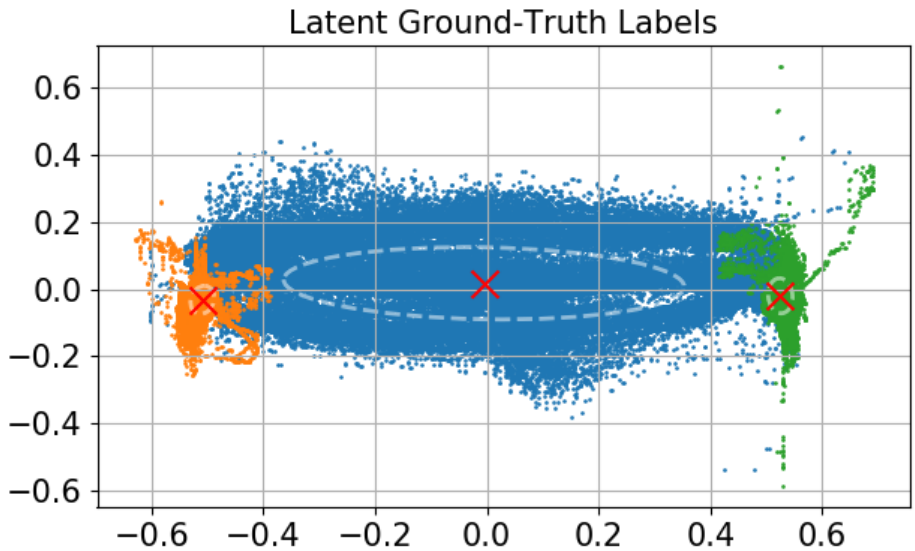

Figure 5.4: Latent space with the Ground-Truth labels, orange indicate the RSS, blue the DS, and green the LSS gait phase. Red crosses and white ellipses are the classes means and covariances respectively.

there exist some points that are incorrectly classified as DS, most points have been correctly classified according to the estimated densities. This is also verified when computing the confusion matrix, shown in Figure 5.8, where we notice that a very small average classification error is achieved, namely 5% for all classes, verifying the accuracy of the proposed framework.

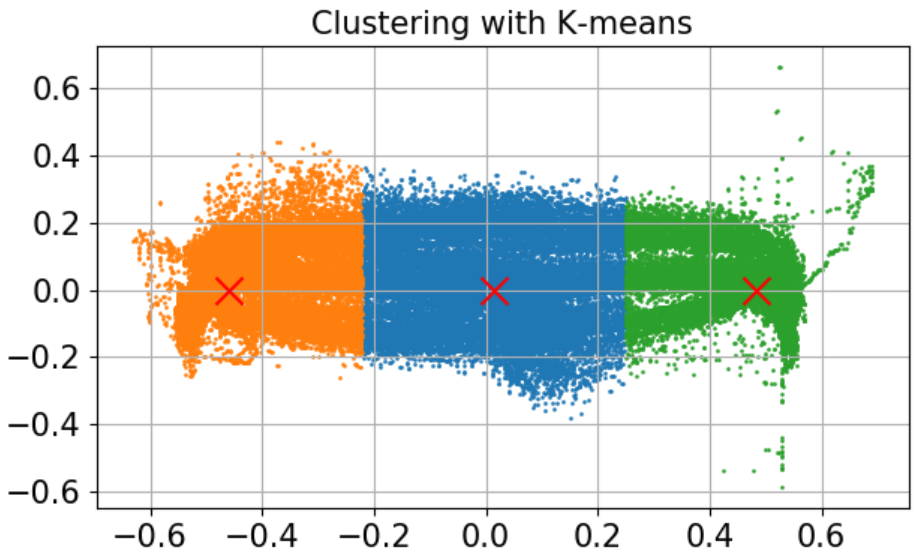

Figure 5.5: Clustering obtained with K-means. Orange indicate the RSS, blue the DS, and green the LSS gait phase. Red crosses are the cluster means.

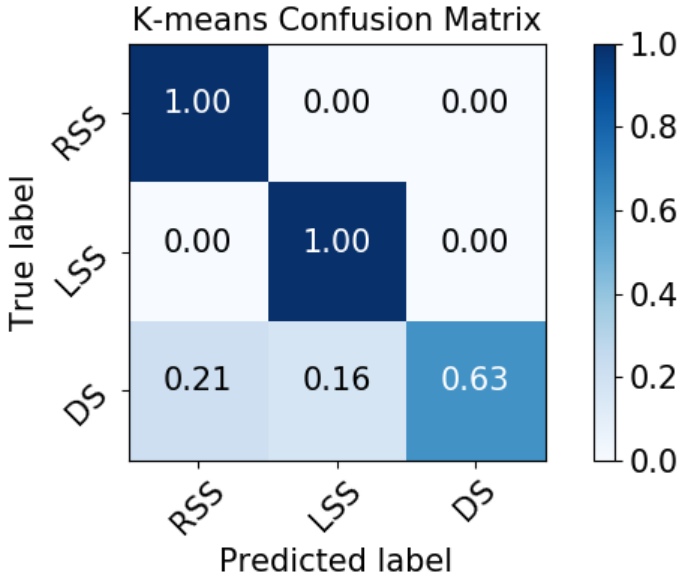

Figure 5.6: K-means confusion matrix. Note the significant error in the DS phase clustering.

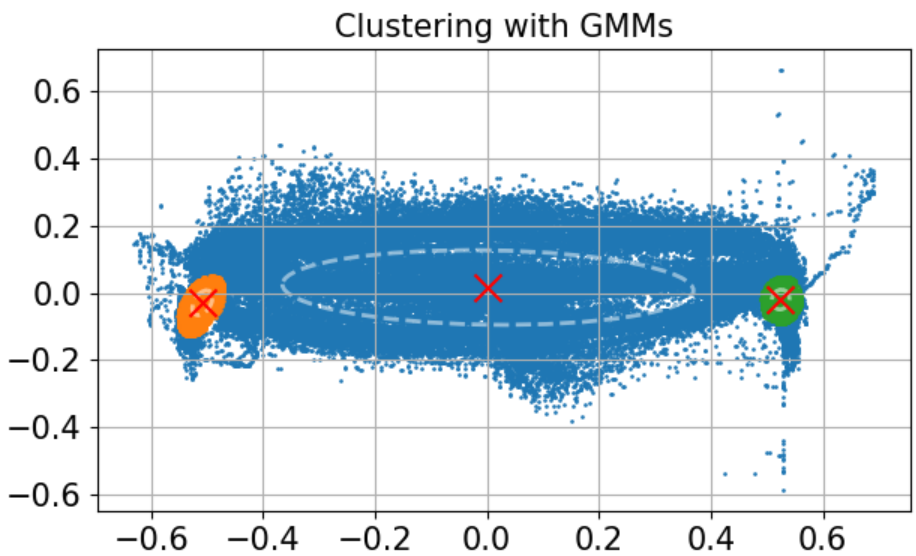

Figure 5.7: Clustering obtained with GMMs. Orange indicate the RSS, blue the DS, and green the LSS gait phase. Red crosses and white ellipses are the cluster means and covariances respectively.

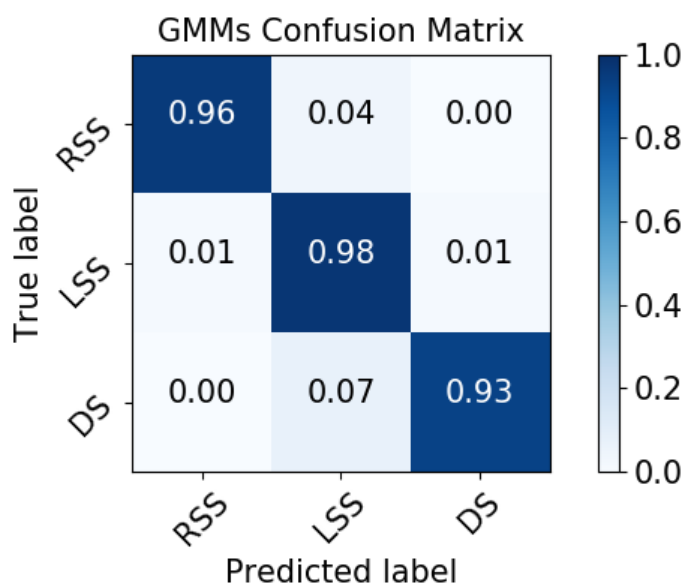

Figure 5.8: GMMs confusion matrix. Clustering is more accurate on average when contrasted to K-means.

Table 5.2: Leg Odometry RMSE.

${}^{w}p_x(m)$	${}^{w}p_y$	${}^{w}p_z$	roll (rad)	pitch	yaw
0.0038	0.0231	0.0199	0.0044	0.0034	0.0064

5.3.2 Application to Leg Odometry

Next, the trained GMMs are employed to predict the gait phase in real-time in order to compute the leg odometry, namely the 3D body position ${}^{w}\boldsymbol{p}_b$ and rotation ${}^{w}\boldsymbol{R}_b$ with respect to the world frame w, for an uneven terrain gait. In this particular gait, Valkyrie climbs up and down a terrain of various friction coefficients.

To compute the leg odometry we initialize the affine transformation linking the current support foot to world frame as ${}^{w}\boldsymbol{T}_s$. Subsequently, when the GMMs predict a single support exchange, we compute the affine transformation from the new support foot to the previous one ${}^{ps}\boldsymbol{T}_s$ with kinematics and update ${}^{w}\boldsymbol{T}_s$ as:

$$ {}^{w}\boldsymbol{T}_s(t) = {}^{w}\boldsymbol{T}_s(t-1){}^{ps}\boldsymbol{T}_s(t) \tag{5.17} $$

where t indicates the discrete-time index.

Leg odometry ${}^{w}\boldsymbol{T}_b$ is computed at each sampling instant as:

$$ {}^{w}\boldsymbol{T}_b(t) = {}^{w}\boldsymbol{T}_s(t){}^{s}\boldsymbol{T}_b(t) \tag{5.18} $$

where ${}^{s}\boldsymbol{T}_b$ is the kinematically computed transformation from the body to the current support foot frame.

In Figure 5.9 the 3D body position and orientation error, when contrasted to the ground-truth trajectories, is illustrated. The conducted gait was cyclic, meaning that Valkyrie was initialized in DS, then went to LSS followed by RSS and the pattern repeats. Thus, for visualization purposes we highlight only the DS phase, which as evident in the previous section is harder to estimate. The final drift was -0.7, 3.3, and -2.0 cm for the position and $1e-5$, $2.5e-5$, and 0.0045 rad for the orientation, respectively for this $76s$ gait. The RMSE for all quantities of interest is shown in Table 5.2. The employed GMMs predicted with 96% accuracy the DS phase, with 89% the LSS and 94% the RSS phase.

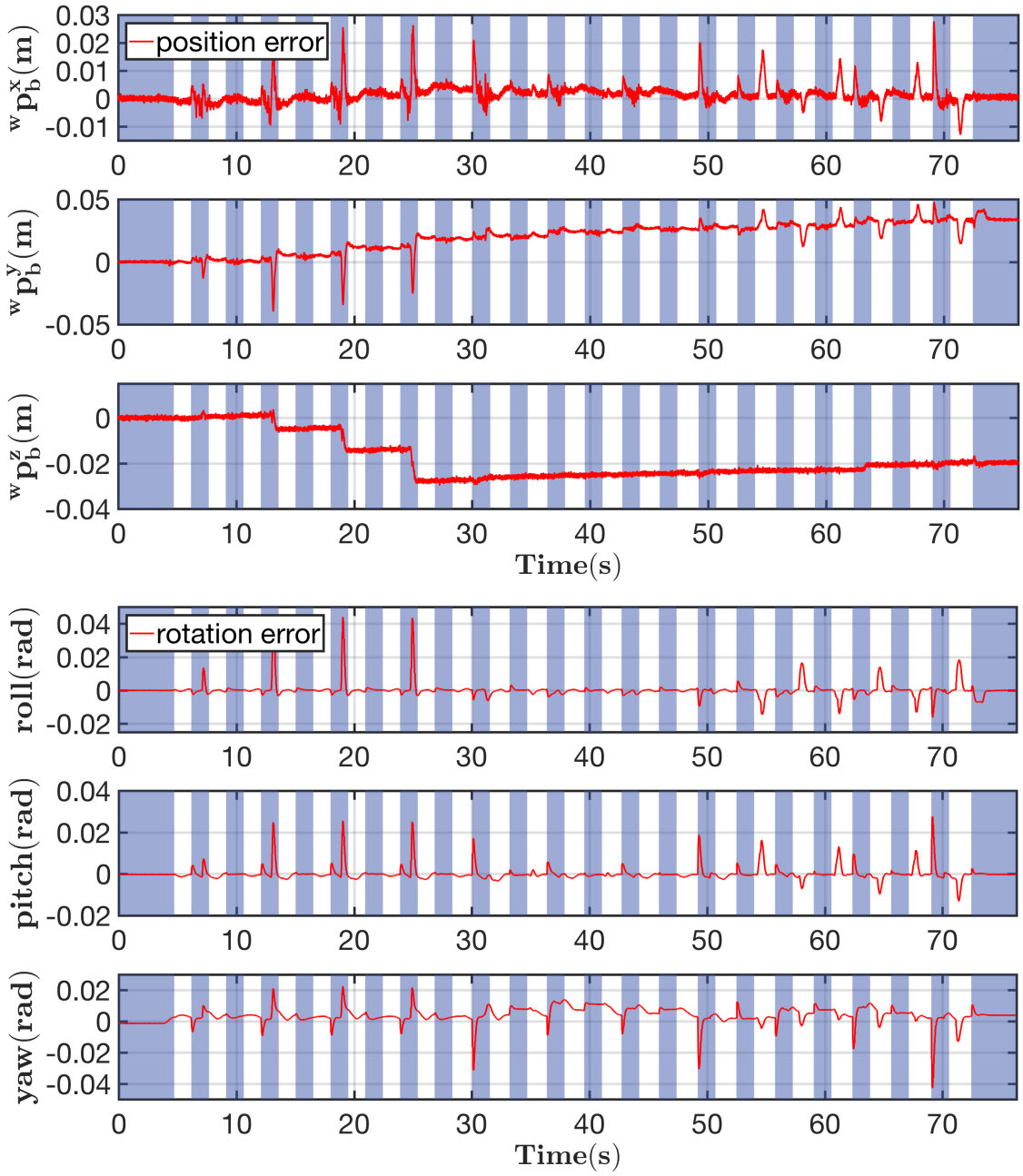

Figure 5.9: **Top**: 3D Body position error, **Bottom**: 3D Body orientation error. Blue regions indicate the estimated DS phases. Since this is a cyclic gait, the first white region corresponds to a LSS phase while the next to a RSS phase and the pattern repeats.

5.3.3 Qualitative Assessment

As evident, the proposed gait phase estimation framework is characterized by accurate performance, which is further preserved when computing leg odometry that can be potentially used in control and/or state estimation.

Additionally, one important remark is that the obtained 2D latent space can appropriately represent the higher dimension gait phase dynamics. To further demonstrate this we executed simple motion primitives e.g. move forward, move backwards, rotate, such primitives are well-captured with specific patterns in the 2D latent space. These observations indicate that locomotion may be a low-dimensional skill, in the sense that a lower dimensional dynamic model can be used for agile and stable omni-directional walking with the high dimensional mechanism. Similar remarks are reported by Westervelt et al. [111] where the full robot dynamics are reduced to a lower dimension manifold called the Hybrid Zero Dynamics and effectively utilized to generate stable walking gaits.

Furthermore, we noticed that K-means does not yield accurate clustering in this particular problem. Nevertheless, when using K-means for binary clustering e.g. LSS or RSS we would have obtained a very accurate classification as also shown in Figure 5.6. This result agrees with [75] where the fuzzy C-means was employed for inferring the contact probability of an end-effector. Consequently, this implies that geometrically the latent-space is not well-separated and can only be clustered when data density is considered.

Finally, GMMs is a probabilistic clustering approach thus, the obtained probabilities can be also interpreted as a measure of uncertainty/quality about the current gait phase.

5.4 Conclusion

In this chapter, we proposed a novel unsupervised learning framework for gait phase estimation and subsequently demonstrated its accuracy when compared to ground-truth data, and efficacy in leg odometry. We demonstrated that a PCA-based 2D latent space is a valid representation of the gait phase dynamics and we established that those dynamics exhibit specific patterns while the robot is omni-directional walking. Such observations point towards locomotion being a low-dimensional skill. Finally, since the proposed framework is based on sensing commonly available on humanoids today, we released GEM [108], an open-source ROS/Python implementation to reinforce further research endeavors.

Chapter 6
Application to other Modules and Platforms

Progress is made by trial and failure; the failures are generally a hundred times more numerous than the successes; yet they are usually left unchronicled.

William Ramsay (1852–1916)

6.1 SEROW in SLAM

Visual Simultaneous Localization and Mapping (visual SLAM) constitutes a challenging task when applied to humanoid robots. While walking, the robot's feet strike the ground and generate sudden accelerations due to rapid and sequential contact switching. These in turn give rise to visual motion blurriness that greatly compromises the performance of the system. In this section we present a dense visual SLAM approach that integrates information from IMU, robot kinematics and contact measurements, to overcome these issues.

Localization is an essential piece of information for autonomous mobile robots, that is usually computed directly using the kinematic information of the platform. For humanoids, however, the kinematics often produces inaccurate estimates since they do not account for the dynamic effects caused by slippage, discontinuous ground contacts and actuation errors. In this context, Visual SLAM can provide an off-the-shelf method to compute the state of a humanoid accurately, using a light and low-cost sensor that is easily mountable on the robot.

However, the motion of a camera mounted on a humanoid has distinctive differences to the motion models assumed in traditional visual SLAM systems. It has a much wider spread, compared to the one for wheeled robots, and follows the oscillating trajectory of the center of mass, as designated by the bipedal gait. This results in blurriness during the

image acquisition process, and reduced performance of the image registration methods.

To solve the aforementioned problems, research focused into new ways of integrating visual SLAM algorithms to humanoid robots [112]. Kagami et al. [113] proposed a vision-based, full-body motion planning method that fuses vision with the localization modules of a humanoid robot. In [114], visual information and a simplified dynamic model of the HRP-2 robot are utilized to improve the robustness of pose tracking. In [115] the authors investigated the performance of a monocular visual SLAM system on the NAO humanoid for which, due to hardware limitations, a robust visual SLAM algorithm was intractable. To overcome those limitations, the authors in [116] mounted an RGB-D sensor to NAO in order to integrate depth information to the robot's footstep planning. In [117], to estimate NAO's pose, a sparse visual SLAM system is employed along with an Extended Kalman Filter, based on kinematic and inertial information. More recently, dense visual SLAM methods are gaining interest, due to resilience to environments with poor features. The authors in [118] developed a dense RGB-D SLAM system for the HRP-4 humanoid, utilizing a reconstruction method suited for dynamic human environments. Finally, in [119] a dense visual SLAM method, with a semi-dense mapping, was embedded on NASA's Valkyrie humanoid.

However, the performance of dense visual SLAM, when computational power and sensor quality are limited, still remains an open question. In the current section we present a robust RGB-D dense SLAM framework, based on KinectFusion [120, 121], that effectively considers the humanoid's kinematics, feet contact status, and IMU measurements to: a) accurately estimate the robot's pose during locomotion, b) construct a dense map of the environment. Our implementation is offered as an open-source ROS C++ and CUDA package at `www.github.com/tavu/kfusion_ros` and achieves real-time execution on an Nvidia Jetson TX2 mounted on a NAO robot.

6.1.1 Method and Results

To efficiently consider the robot's kinematics and contact effects we employ the State Estimation Robot Walking (SEROW) framework [47, 49, 76]. The latter fuses IMU, joint encoder, and Force/Torque (F/T) measurements to accurately estimate the following state vector $\boldsymbol{x}_t$:

$$\boldsymbol{x}_t = \begin{bmatrix} {}^b\boldsymbol{v}_b & {}^w\boldsymbol{R}_b & {}^w\boldsymbol{p}_b & \boldsymbol{b}_\omega & \boldsymbol{b}_a \end{bmatrix}^\top$$

where ${}^w\boldsymbol{p}_b$, ${}^w\boldsymbol{R}_b$ denote the base position and rotation with respect to a world frame w, ${}^b\boldsymbol{v}_b$ is the linear velocity, and $\boldsymbol{b}_\omega$, $\boldsymbol{b}_\alpha$ are IMU biases, in the base frame b.

The kinematic information derived from SEROW are used to improve the performance of the localization and mapping processes in visual SLAM. To accomplish this we modify

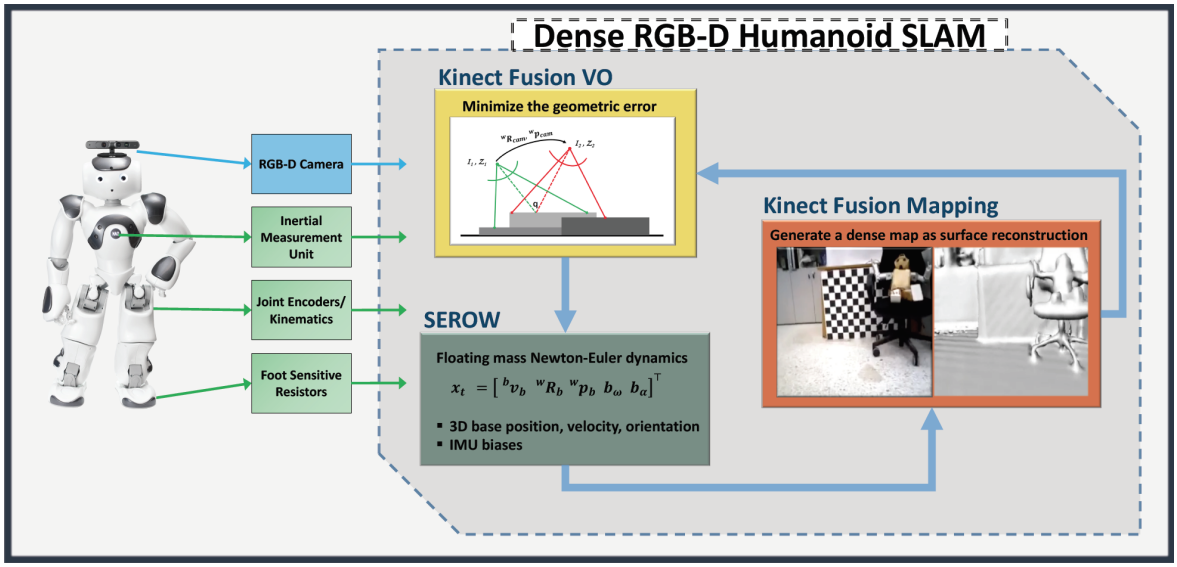

Figure 6.1: Dense RGB-D Humanoid SLAM architecture. The framework utilizes RGB-D images along with IMU, encoder, and F/T measurements in an consistent loop.

KinectFusion [120] and separate two functionalities: a) Estimate the robot's pose using VO and b) Generate a dense map of the environment and rectify pose estimates using this map. In such it is straightforward to add SEROW in the loop, as illustrated in Fig. 6.1. SEROW incorporates the VO to estimate a more accurate robot pose that is then propagated to the mapping module.

We evaluate our system by executing it on a NAO humanoid, while navigating in an office environment (Fig. 6.2 top). Figure 6.3 demonstrates the corresponding 3D base position (Fig. 6.3, top) and orientation (Fig. 6.3, bottom) as estimated by KinectFusion, kinematics and KinectFusion with SEROW respectively. As it is shown, the estimates provided by KinectFusion alone exhibit large errors in the vertical axis, and are subject to drift. The latter is further illustrated in Table 6.1 that presents the final pose drift for the three implementations. Evidently, the fusion of KinectFusion with SEROW yields accurate estimates for all quantities, and improves the quality of the obtained map, as depicted in Fig. 6.2 bottom. The conducted experiment is available in HD quality at `https://youtu.be/mLdNwHl9cgo`. Our implementation achieved 29Hz on the Jetson TX2 module, given that RGB-D images are available at 30fps and volumetric rendering is disabled.

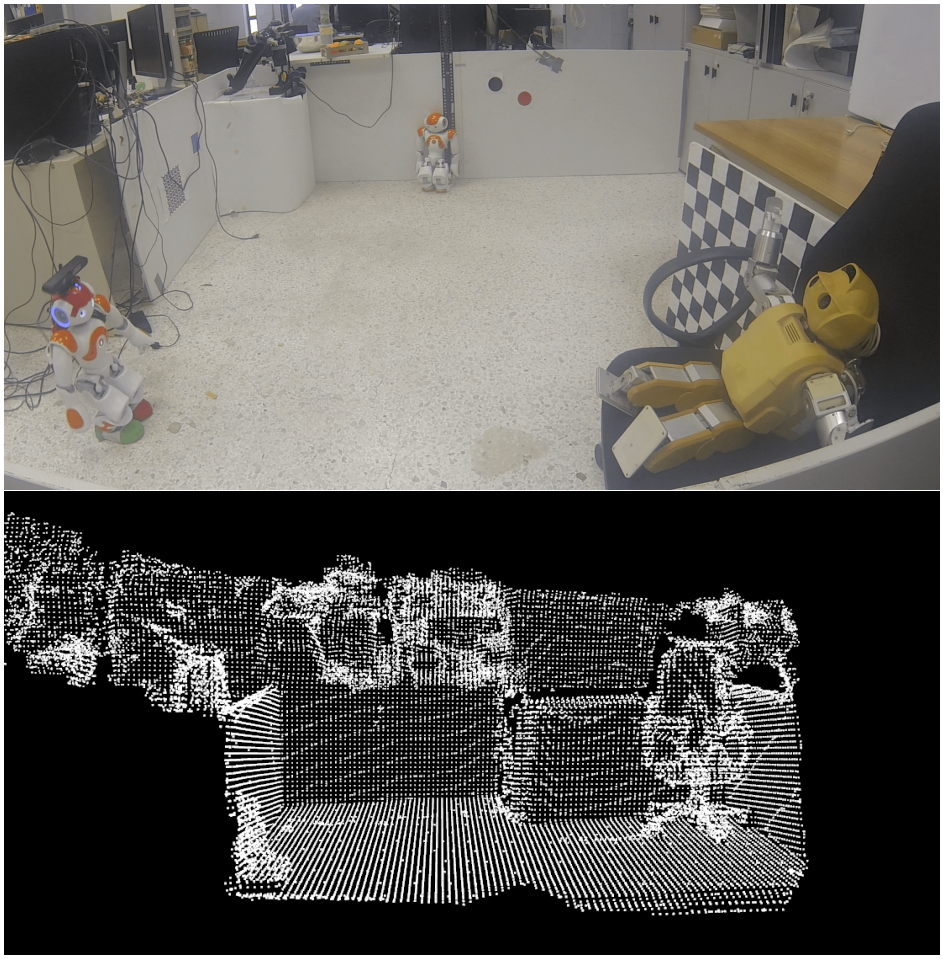

Figure 6.2: **Top**: NAO navigating in an office environment. **Bottom**: Reconstructed 3D map.

Table 6.1: Pose Drift

	${}^{w}p_x(m)$	${}^{w}p_y$	${}^{w}p_z$	roll (deg)	pitch	yaw
KF	0.025	0.047	0.287	2.80	2.35	2.86
Kin	0.261	0.331	0.023	6.5e-4	0.65	7.01
KF+S	0.011	0.038	0.031	6.5e-4	0.65	3.69

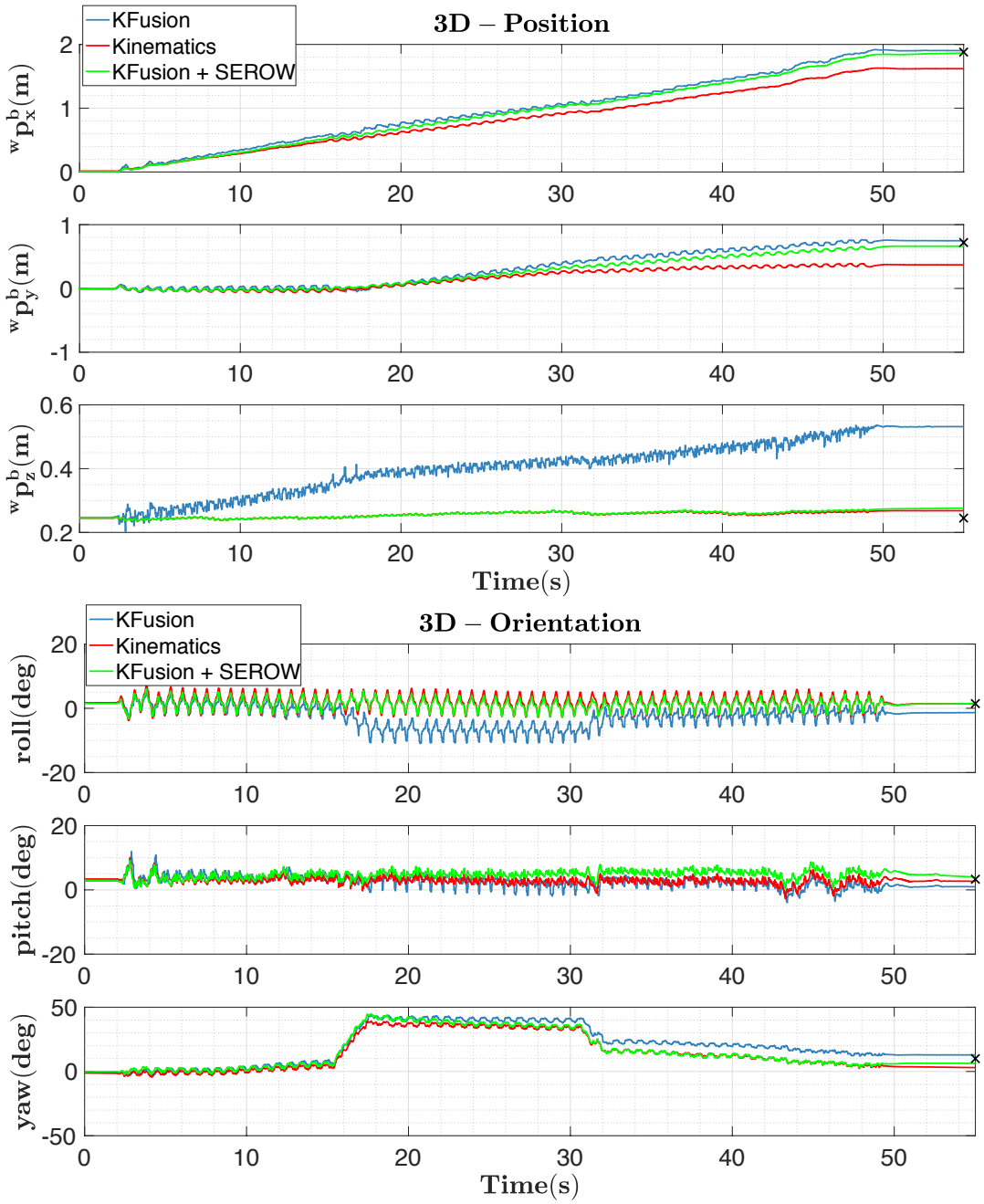

Figure 6.3: Pose estimates generated by the three algorithms. **Top**: 3D body position. **Bottom**: 3D body orientation. Blue lines indicate the estimates from VO, red lines from the kinematics, green lines from KinectFusion and SEROW. Black crosses are the measured final states.

6.1.2 Conclusion

In this section, we presented a method to fuse a dense visual SLAM algorithm with the kinematic information from a humanoid robot. Our framework exhibits increased performance when compared to a solely RGB-D SLAM and can achieve real-time execution on an embedded GPU device.

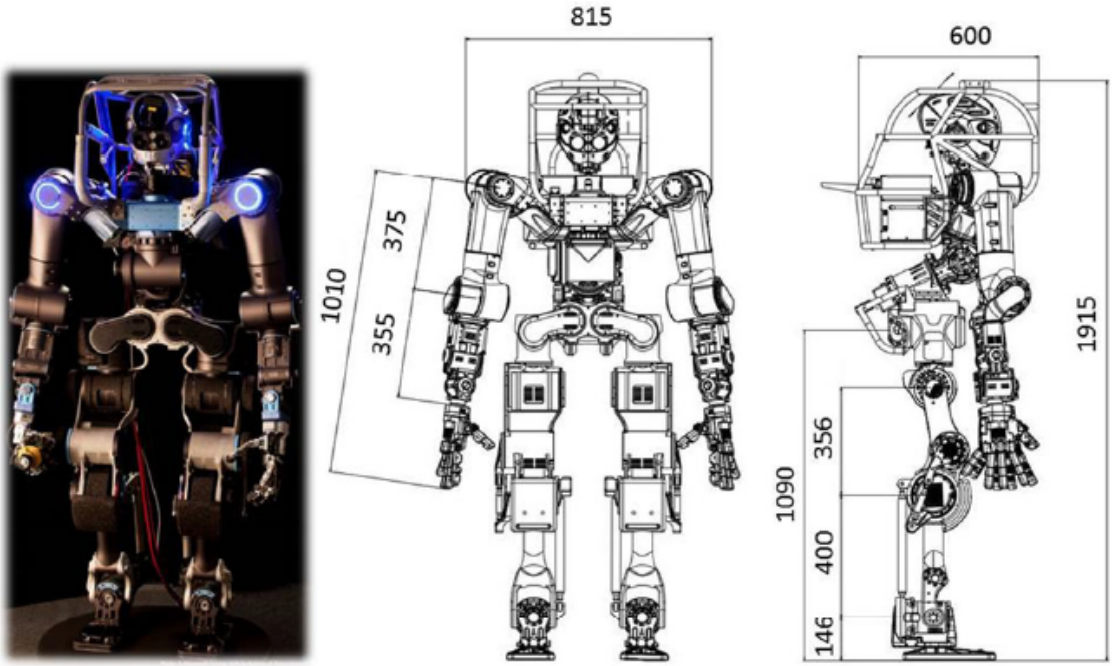

Figure 6.4: WALK-MAN humanoid. All dimensions are specified in mm. [122]

6.2 SEROW integrated with WALK-MAN

WALK-MAN [122–124] is a humanoid robot designed and constructed by the Instituto Italiano di Tecnhologia (IIT) within the context of the European WALK-MAN integrated project, funded in 2013. The robot has dimensions similar to the dimensions of an adult human, as evident in Figure 6.4. More specifically, its height from the sole of its foot to the top of its head is 1.915m. The shoulder width is 0.815m while its depth at the torso is 0.6 m. The total weight of WALK-MAN is 132kg of which 14kg is the mass of the power pack and 7kg is the mass of the protection roll bar structure around the torso and head.

WALK-MAN's upper body (not considering hands and neck) has 17 DoF; each arm has 7 DoF and the trunk has a 3 DoF waist. WALK-MAN arm kinematics closely resembles an anthropomorphic arrangement with 3 DoF at the shoulder, 1 DoF at the elbow, 1 DoF for the forearm rotation and 2 DoF at the wrist.

WALK-MAN is employed with two absolute magnetic encoders in every actuator unit for accurately measuring the joint states. Joint torque sensing is implemented using an elastic torsion bar whose deflection is measured by two high resolution absolute encoders as in position sensing. Subsequently, to derive the joint torque measurement the stiffness of the bar is used which is accurately determined in the calibration phase. Furthermore, WALK-MAN has four 6D F/T sensors located at the wrist and ankle joints that can measure

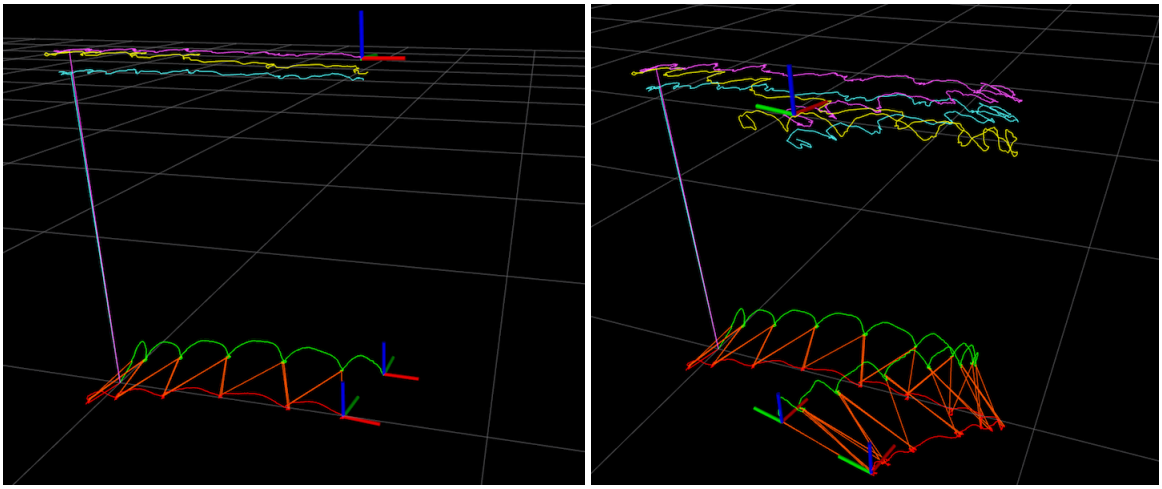

Figure 6.5: SEROW in ROS rviz. In top, the yellow line indicates the Ground-Truth, purple is the base trajectory, green is the base leg odometry, and teal is the CoM trajectory. In bottom, green and red lines indicate the left and right feet trajectory, while the orange line is the support foot trajectory.

the contact wrenches the robot experiences with the environment.

WALK-MAN also has a VectorNav VN-100 IMU at the pelvis link and incorporates a Multisense M7 sensor at the head to provide a stereo vision system with an integrated FPGA unit, an IMU, and a LIDAR. To facilitate vision-based applications an i7 quad core processor COM express PC has been installed in the back side of the head. Additionally, another COM express PC based on an i7 quad core processor exists in the torso to facilitate motion planning and real-time gait control.

Footstep planning on various type of terrains [125] is accomplished according to [126] based on visual feedback [127]. Walking pattern generation is achieved as in [102, 103] through the XBotCore [104] middleware and stabilization is performed with the OpenSoT [105] whole-body controller every 500Hz. Desired joint positions are realized with PID position control, implemented in a distributed embedded electronic system with one board per joint running at 1kHz.

SEROW is implemented on the robot and achieves real-time execution with a 200 Hz cycle, which is the rate the IMU provides new measurements through ROS. State estimation with kinematic-inertial-F/T data is illustrated on Figure 6.5 for a dynamic omnidirectional gait. The ground-truth base pose was recorded with an OptiTrack motion capture system at 120 Hz. Figure 6.6 depicts the 3D-base position and orientation as estimated with SEROW and with a contact based EKF [56]. Notice, both filters exhibit large errors in the y-axis position, this is due to drift. Commonly while walking, the robot hits the ground and generates enormous GRFs that often cause the feet to bounce back and slip. These

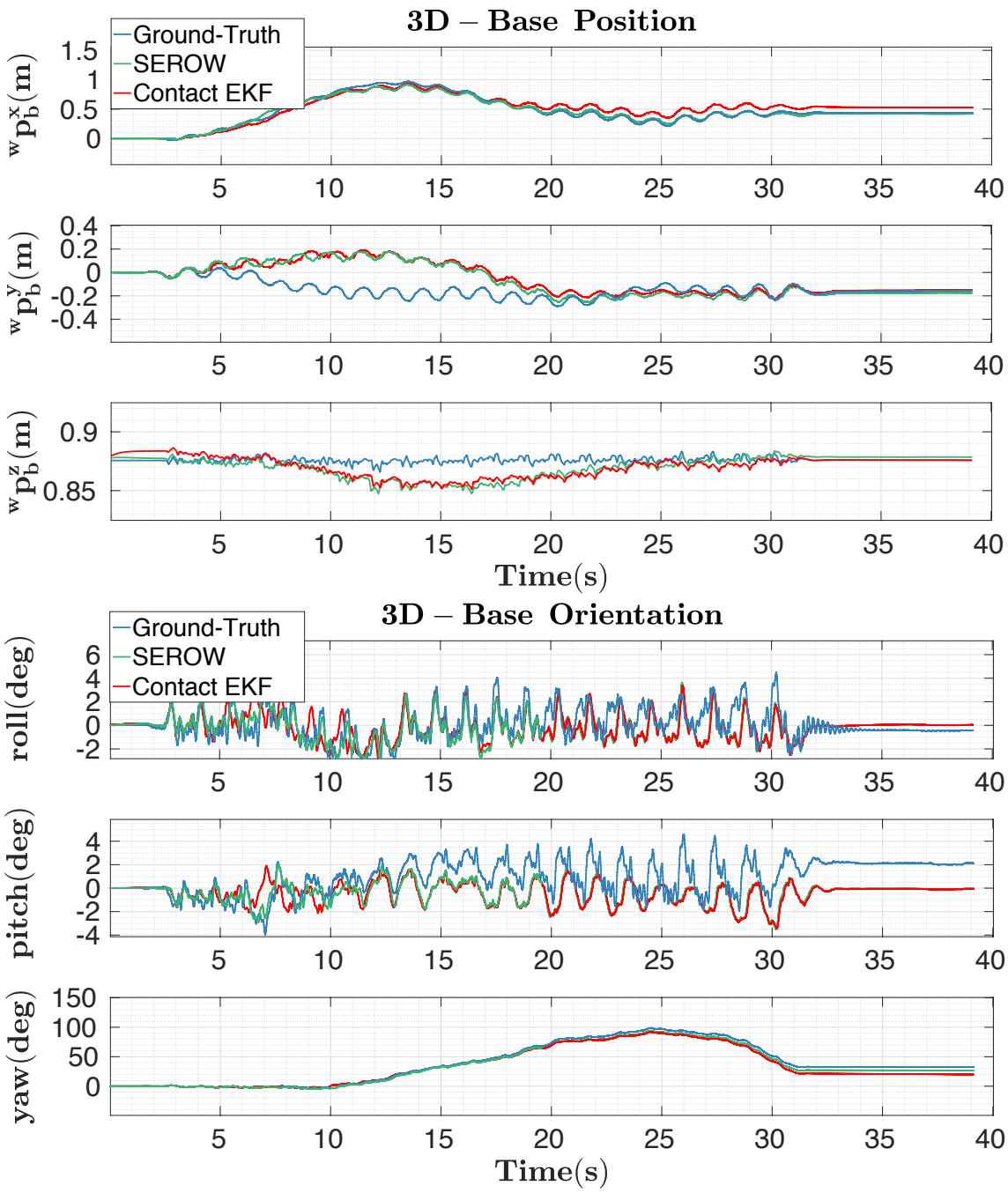

Figure 6.6: **Top:** 3D-Base position, **Bottom:** 3D-Base orientation. Blue lines indicate the Ground-Truth trajectories, green lines are the estimated trajectories with SEROW and red lines the estimated trajectories with a contact-based EKF.

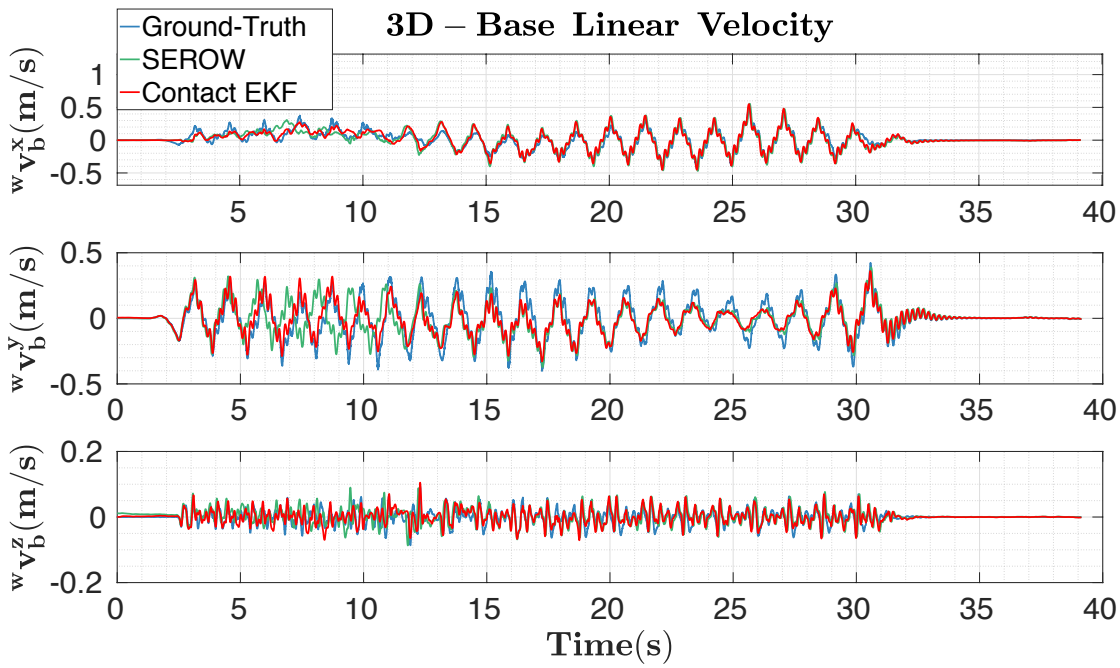

Figure 6.7: 3D-Base velocity. Blue lines indicate the Ground-Truth trajectories, green lines are the estimated trajectories with SEROW and red lines the estimated trajectories with a contact-based EKF.

errors are also evident in the estimated 3D-base linear velocities shown in Figure 6.7. Although both filters demonstrated similar accuracy, SEROW was marginally more accurate.

Subsequently, the 3D-CoM position and velocity was estimated with SEROW. Since the CoM ground-truth was not available, we employed the base pose as estimated by the contact based EKF and the kinematically computed CoM for comparison. Figure 6.8 illustrates the 3D-CoM position and velocity as estimated by the two schemes. Notice, once again base estimation drift propagates to CoM estimation. Furthermore, in the vertical axis the CoM as estimated with SEROW exhibits a more dynamic behavior that the kinematically computed CoM, probably due to the feet impacting the ground.

State estimation incorporating additionally external VO has been already demonstrated in section 4.3.

Accordingly, GEM was employed to derive the gait phase, as presented in chapter 5. The dataset had 8069 entries:

$$\mathcal{D} = \left\{ {}^{b}\delta \boldsymbol{c}_n, \delta\theta_n^{\text{roll}}, \delta\theta_n^{\text{pitch}}, \Delta \boldsymbol{f}, \Delta \boldsymbol{\tau} \right\} \in \mathbb{R}^{11\times 8069} \tag{6.1}$$

Contact ground-truth data were not available. Dimensionality reduction to two dimensions was performed with both PCA and autoencoders. PCA yielded a reprojection error of

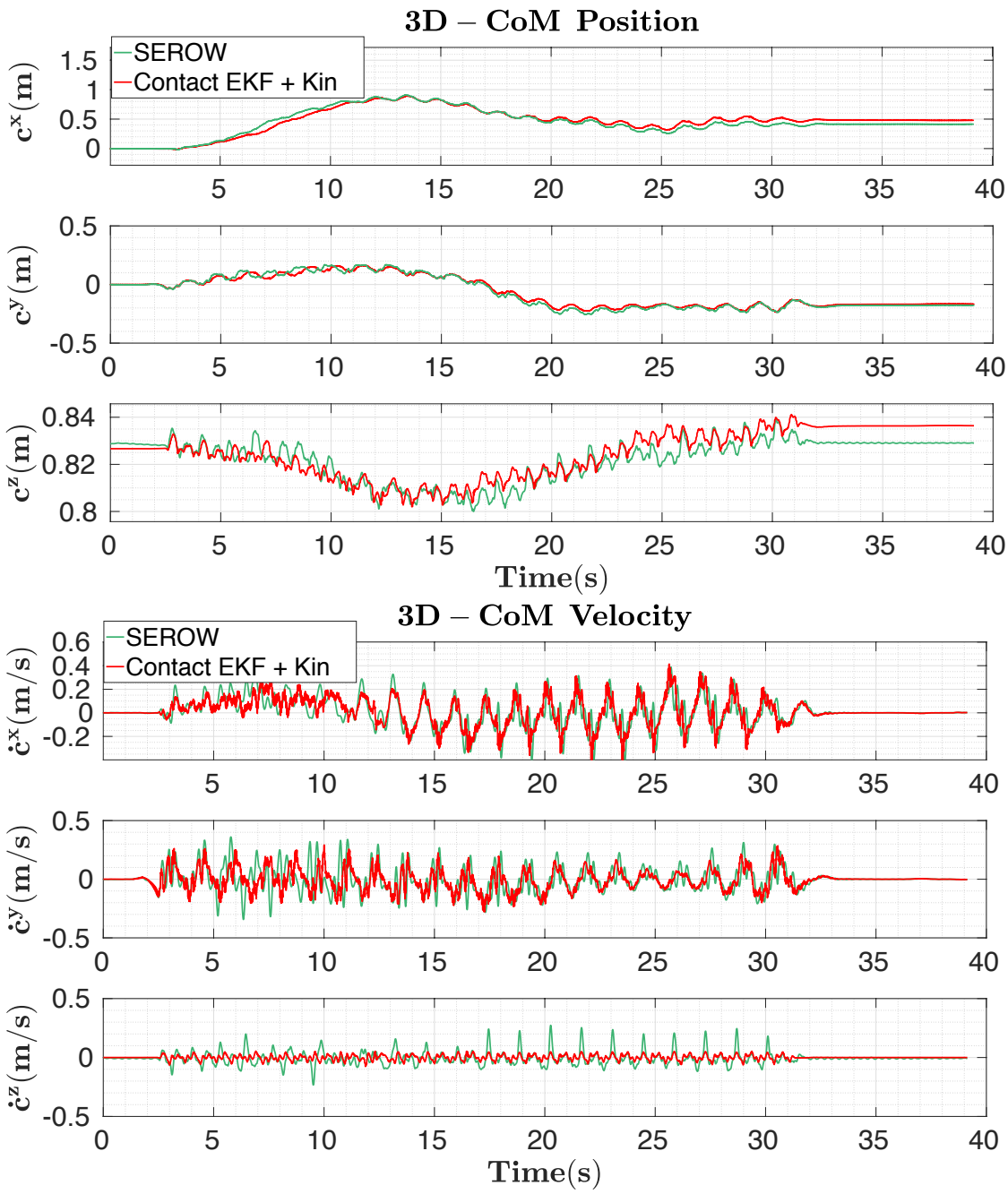

Figure 6.8: **Top:** 3D-CoM position, **Bottom:** 3D-CoM velocity. Green lines indicate the estimated trajectories with SEROW and red lines the estimated trajectories with a contact-based EKF and kinematics.

0.0070 while autoencoders achieved reprojection error of 0.0064. As in section 5.3 the de-

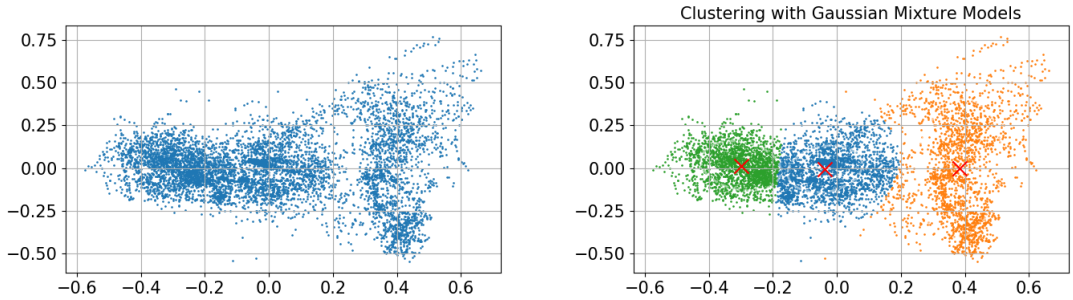

Figure 6.9: **Left:** Latent space with PCA, **Right:** Clusters obtained with GMMs, orange indicates the LSS, blue the DS, and green the RSS, crosses indicate the cluster centers.

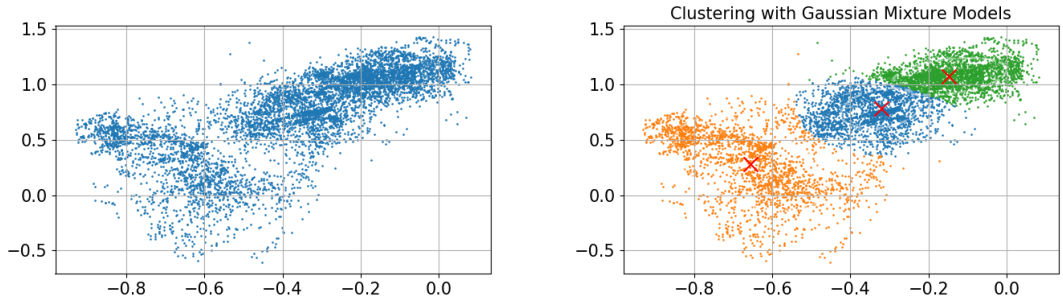

Figure 6.10: **Left:** Latent space with autoencoders, **Right:** Clusters obtained with GMMs, orange indicates the LSS, blue the DS, and green the RSS, crosses indicate the cluster centers.

rived latent spaces are similar in shape. More specifically, the PCA latent space (Figure 6.9 left) is a rotated/scaled version of the latent space obtained with the autoencoders (Figure 6.10 left). Subsequently, GMMs are employed for clustering. Once again, three dense clusters are obtained in both cases as depicted in Figure 6.9 right and Figure 6.10 right. Interestingly, this latent space does not sufficiently represent the higher dimensional space. To this end, we expect the estimated gait phase to not be very accurate. This is evident from the explained variance of the two principal components obtained with the PCA. The first principal component has an explained variance of 0.3936% while the second explains another 0.1621% of the data variance. This suggests that more dimensions are needed to sufficiently represent the $11D$ dataset.

The major difference from the experiments conducted with the Valkyrie robot in section 5.3 is that Valkyrie performed slow pace statically stable walking while WALK-MAN achieved faster and more dynamic gaits. Thus, an important remark is that more fea-

tures are needed to derive an accurate gait phase while dynamic walking. To this end, approaches that rely only on the vertical GRF to determine the feet contact status are not suitable for agile dynamic walking.

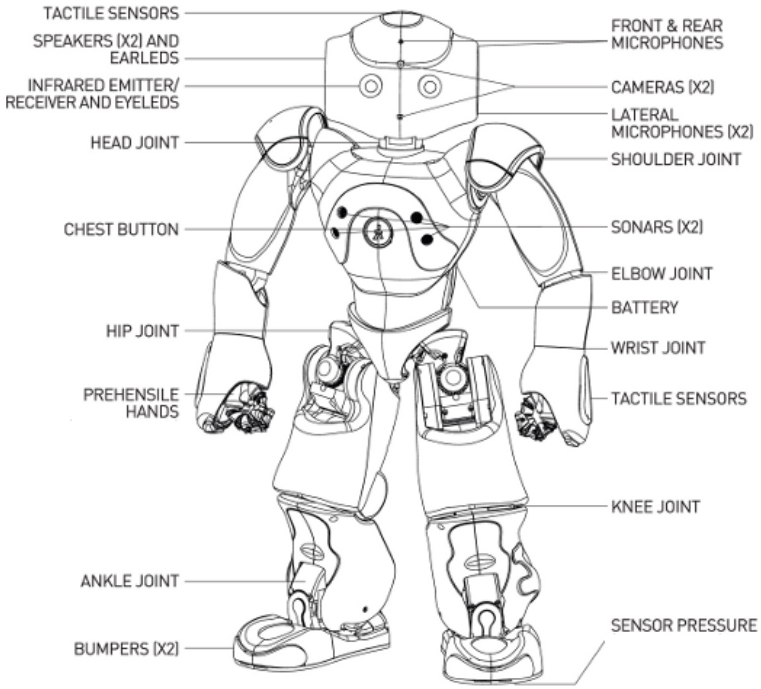

Figure 6.11: Aldebaran NAO v4.0 components

6.3 SEROW integrated with NAO

The hardware platform that is currently used for the RoboCup Standard Platform League (SPL) is NAO, an integrated, programmable, medium-sized humanoid robot originally developed by Aldebaran Robotics in Paris, France and today owned by Softbank robotics. The robot's development began with the launch of Project NAO [128] in 2004. In August 2007, NAO officially replaced Sony's Aibo quadruped robot in the RoboCup SPL. In the past few years NAO has evolved over several designs and several versions up to the very recent version 6.0.

NAO (version 4.0), shown in Figure 6.11, is a 58cm, 5.18kg humanoid robot. The NAO robot carries a fully capable computer on-board with an ATOM Z530 processor at 1.6GHz, 1GB SDRAM, and 2 GB flash memory running an Embedded Linux distribution. It is powered by a 6-cell Lithium-Ion battery which provides about 30 minutes of continuous operation and communicates with remote computers via an IEEE 802.11g wireless or a wired ethernet link.

NAO has 25 degrees of freedom; 2 in the head, 6 in each arm, 5 in each leg and 1 in the pelvis (there are two pelvis joints which are coupled together on one servo and cannot move independently). NAO, also, features a variety of sensors. Two cameras are mounted on the head in vertical alignment providing non-overlapping views of the lower and dis-

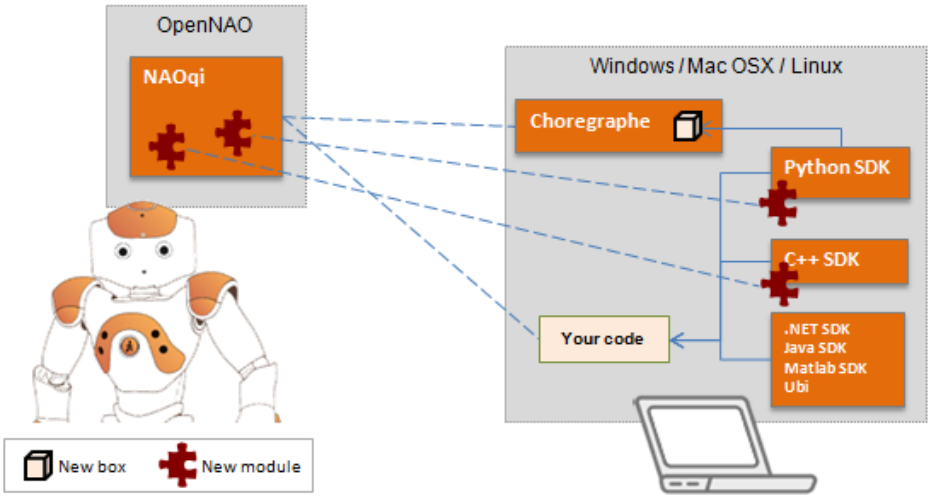

Figure 6.12: Embedded and desktop software for the NAO robot

tant frontal areas, but only one is active each time and the view can be switched from one to the other almost instantaneously. Each camera is a 640×480 VGA devise operating at 30fps. Four sonars (two emitters and two receivers) on the chest allow NAO to sense obstacles in front of it. In addition, NAO has a rich inertial unit, with one 2-axis gyroscope and one 3-axis accelerometer, in the torso that provides real-time information about its instantaneous body movements. Two bumpers located at the tip of each foot are simple ON/OFF switches and can provide information on collisions of the feet with obstacles. Finally, an array of force sensitive resistors on each foot delivers feedback of the forces applied to the feet, while encoders on all servos record the actual values of all joints at each time.

Aldebaran Robotics has equipped NAO with both embedded and desktop software [129] to be used as a base for further development (Figure 6.12). The embedded software, running on the motherboard located in the head of the robot, includes an embedded GNU / Linux distribution and NAOqi, the main proprietary software that runs on the robot and controls it. NAO's desktop software includes Choregraphe, a visual programming application which allows the creation and the simulation of animations and behaviors for the robot before the final upload to the real NAO, and Telepathe which provides elementary feedback about the robot's hardware and a simple interface to accessing its camera settings.

As far as the NAOqi framework is concerned, it is cross-platform, cross-language, and provides introspection which means that the framework knows which functions are available in the different modules and where. It provides parallelism, resources, synchronization, and events. NAOqi, also, allows homogeneous communication between different modules (motion, audio, video), homogeneous programming, and homogeneous infor-

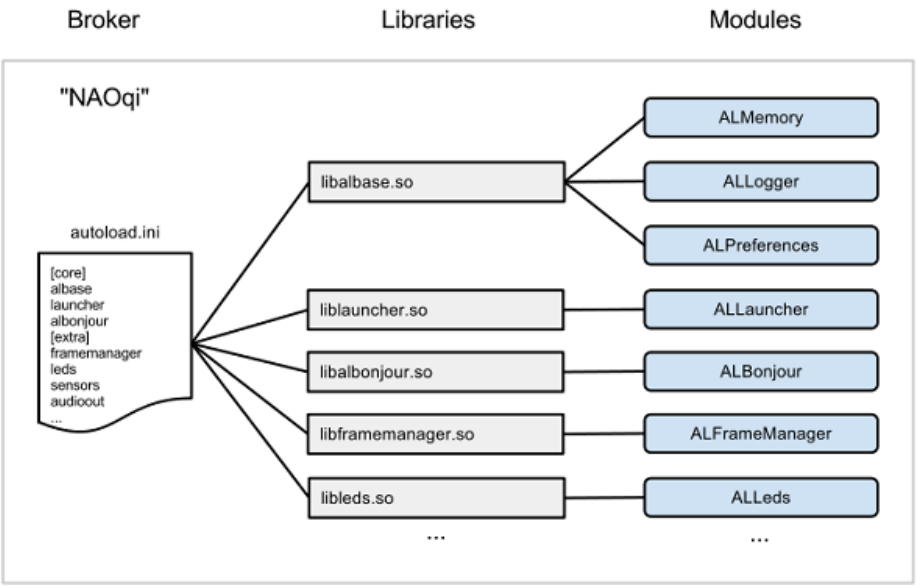

Figure 6.13: The NAOqi process

mation sharing. Software can be developed in C++, Python, and Urbi. The programmer can state which libraries have to be loaded when NAOqi starts via a preference file called `autoload.ini`. The available libraries contain one or more modules, which are typically classes within the library and each module consists of multiple methods (Figure 6.13).

Recently, we developed an omni-directional walking engine for the NAO robot based on MPC with ZMP constraints and the Cart and Table model [67]. The latter achieved fast stable bipedal walking on flat terrain and was employed in the context of Robocup by the Greek Kouretes team [130]. Feedback was available from FSR and kinematics. In this section, we propose a 3D omni-directional walk engine that naturally encapsulates SEROW and achieves 3D walking with step location and step duration adjustment. Our implementation achieves real-time execution on NAO's $10ms$ control cycle, is open-source as a NAOqi module with a ROS C/C++ wrapper [131]. The latter are publicly available at `www.github.com/mrsp/nao_walk`.

The proposed module consists of three parts: a) the footstep planner that provides with desired footstep locations, b) the motion planner, which generates stable walking patterns, c) posture controllers, that stabilize the walking patterns. In all parts SEROW provides the mandatory body and feet affine transformations, as well as the 3D-CoM position, velocity, and external forces acting on the CoM.

The footstep planner is based on [133, 134], which efficiently generates 2D step plans for humanoid robot navigation with anytime repairing A^* (ARA^*) and/or randomized A^* (R^*) search. This framework has been extended to 3D by Stumpf et al. [135] to prop-

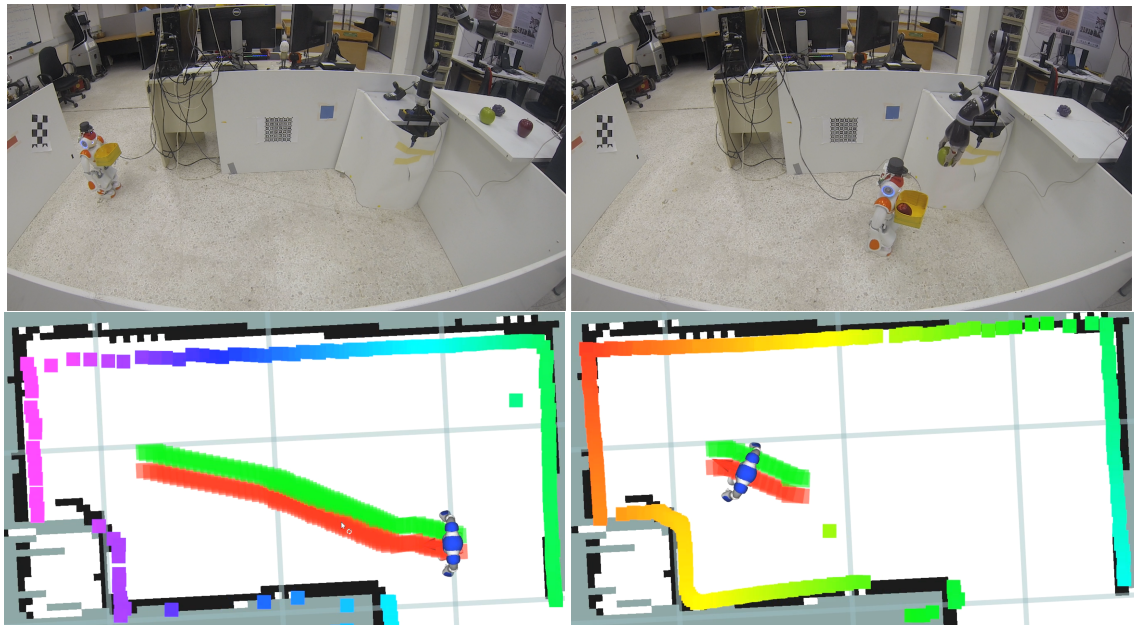

Figure 6.14: NAO navigation in a lab environment. Dynamic footstep planning allows NAO to reach the workspace of a JACO manipulator [132] in order to obtain some fruits. Green/Red steps indicate the right/left leg reference footsteps respectively, while color cubes are the LIDAR measurements.

erly consider the perceived world during step planning. In this work, we employ an RP-LIDAR360° on NAO's head and utilize SEROW with the ROS gmapping [136] package to perform SLAM. The generated 2D map is then used for dynamic footstep planning along with the ROS amcl package [137], as shown in Figure 6.14. In this experiment, NAO navigates in a lab environment to reach the workspace of a JACO manipulator [132]. When this happens, JACO detects a bowl mounted on NAO and fetches some fruits.

Subsequently, the generated footstep plan is propagated to the walk engine for execution. The complete walk engine is illustrated in Figure 6.15, where as evident SEROW provides feedback for both the motion planning and the real-time gait stabilization module.

At the beginning of each SS phase, 2D-step location and timing adjustment is performed as in [138]. To this end, the following QP is solved to optimally determine the desired step location and duration that achieve zero-step capturability [21]:

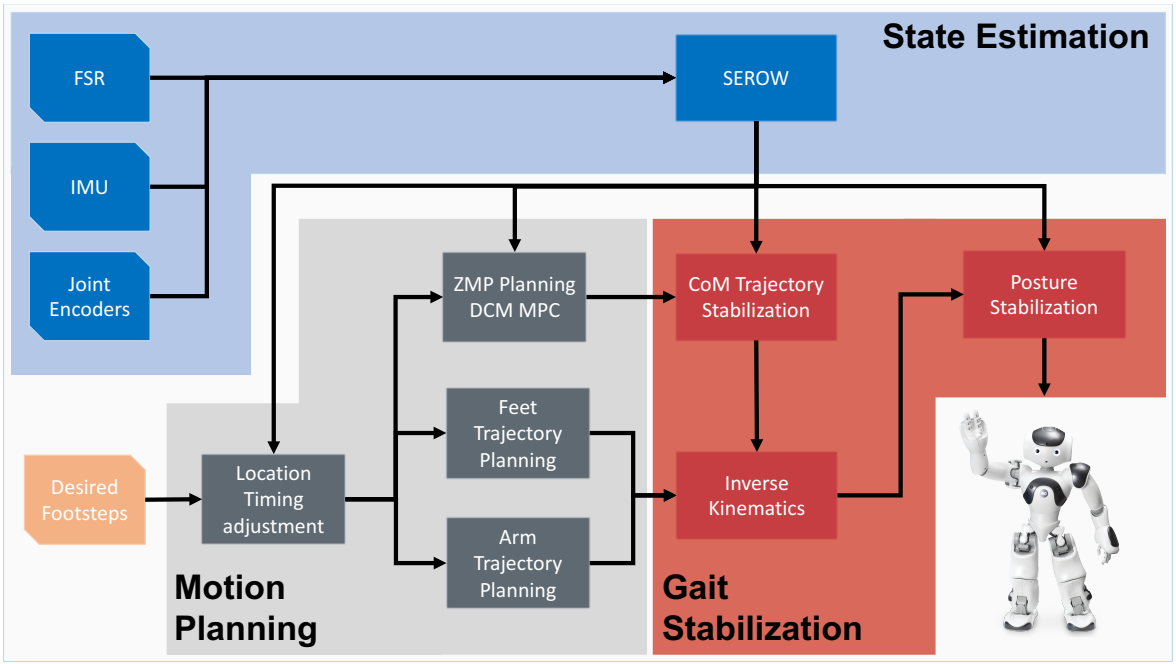

Figure 6.15: NAO Walk Engine consisting of: a) the state estimation module, b) the motion planning module, and c) the gait stabilization module.

$$\min_{\boldsymbol{u}_T,\tau,\boldsymbol{b}} \quad \alpha_1\|\boldsymbol{u}_T - \boldsymbol{u}_0 - \boldsymbol{L}^x_{\text{nom}}\|^2 + \alpha_2\|\tau - \tau_{\text{nom}}\|^2 + \alpha_3\|\boldsymbol{b} - \boldsymbol{b}_{\text{nom}}\|^2 \tag{6.2a}$$

$$\text{subject to} \quad e^{\omega T_{\min}} \leq \tau \leq e^{\omega T_{\max}}, \tag{6.2b}$$

$$\boldsymbol{L}_{\min} \leq {}^{w}\boldsymbol{R}_s^{\top}(\boldsymbol{u}_T - \boldsymbol{u}_0) \leq \boldsymbol{L}_{\max}. \tag{6.2c}$$

$$\boldsymbol{u}_T - (\boldsymbol{\xi}_0 - \boldsymbol{p}_0)\tau + \boldsymbol{b} = \boldsymbol{p}_0 \tag{6.2d}$$

where α_* are optimization weights, $\boldsymbol{u}_0$, $\boldsymbol{u}_T$ are the support and swing foot position, $\boldsymbol{L}_{\text{nom}}$ is the nomimal displacement, τ and τ_{nom} are the exponential step and nomimal step duration, $\boldsymbol{b}$, $\boldsymbol{b}_{\text{nom}}$ are the DCM and nomimal DCM offset at the end of the step, and ${}^{w}\boldsymbol{R}_s$ is the support foot rotation. The only feedback required is the initial COP $\boldsymbol{p}_0$ and the DCM position $\boldsymbol{\xi}_0$ as estimated by SEROW. Consequently, step duration can be obtained as:

$$T = \frac{1}{\omega}\log\tau \tag{6.3}$$

This QP is five dimensional and is efficiently solved on NAO with an active set convex optimization algorithm. The DCM offset $\boldsymbol{b}$ is directly related to the stability of the robot. To this end, in order to obtain stable gaits α_3 should be larger than the other weights. Never-

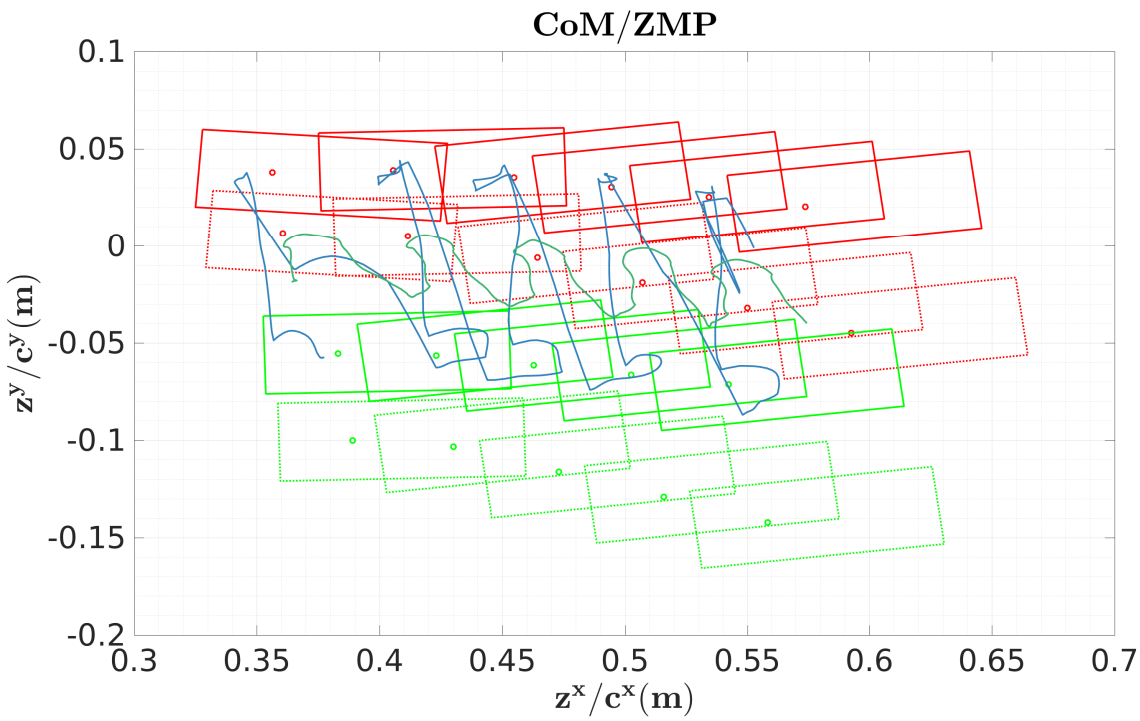

Figure 6.16: Step adjustment in the 2D plane. Red/Green dotted lines indicate the planned steps, while red/green solid lines are the adjusted steps. The blue line is the measured ZMP with the FSRs and the dark green line is the CoM as estimated with SEROW.

theless, it is often the case the desired DCM offset cannot be realized. In the latter scenario, if the robot is capturable [21], multiple steps are required to converge back to the nomimal gait. A consecutive step adjustment to a reference footstep plan is shown in Figure 6.16, where also the ZMP as measured with the NAO's FSRs and the CoM as estimated with SEROW is depicted. Subsequently, the reference ZMP, swing foot and swing arm trajectories are generated with cubic splines.

In order to generate stable walking patterns we formulate a novel MPC scheme based on the DCM. The CoM and the DCM dynamics are described by the following first order differential equations [20]:

$$\dot{\boldsymbol{\xi}} = \omega(\boldsymbol{\xi} - \boldsymbol{p}) \tag{6.4}$$

$$\dot{\boldsymbol{c}} = \omega(\boldsymbol{\xi} - \boldsymbol{c}) \tag{6.5}$$

Notice, the CoM dynamics are linear independent from the DCM dynamics and furthermore stable in the sense that CoM follows the DCM. To this end, it suffices to stabilize only

the unstable DCM dynamics to obtain stable CoM walking patterns. Consider the x-axis for simplicity, although the same relations hold for the dynamics in y and z axes. Let the input u be the derivative of the ZMP $\dot{z}^x$. Then, the state-space can be formulated as:

$$\frac{d}{dt}\begin{bmatrix}\xi^x\\ z^x\end{bmatrix} = \overbrace{\begin{bmatrix}\omega & -\omega\\ 0 & 0\end{bmatrix}}^{\boldsymbol{A}}\overbrace{\begin{bmatrix}\xi^x\\ z^x\end{bmatrix}}^{\boldsymbol{x}} + \overbrace{\begin{bmatrix}0\\ 1\end{bmatrix}}^{\boldsymbol{B}} u \tag{6.6}$$

$$y = \overbrace{\begin{bmatrix}0 & 1\end{bmatrix}}^{\boldsymbol{C}} \boldsymbol{x} \tag{6.7}$$

Additionally, consider the embedded integrator state x_e:

$$\boldsymbol{x}^e = \begin{bmatrix}\Delta\xi^x\\ \Delta z^x\\ z^x\end{bmatrix} \tag{6.8}$$

The embedded integrator has been demonstrated to eliminate steady state errors since an integral term is added to the control law [67]. The discrete-time state-space with embedded integrator is given by:

$$\boldsymbol{x}^e_{k+1} = \overbrace{\begin{bmatrix}\boldsymbol{A}_d & 0\\ \boldsymbol{C}_d\boldsymbol{A}_d & 1\end{bmatrix}}^{\boldsymbol{A}_e}\boldsymbol{x}^e_k + \overbrace{\begin{bmatrix}\boldsymbol{B}_d\\ \boldsymbol{C}_d\boldsymbol{B}_d\end{bmatrix}}^{\boldsymbol{B}_e}\Delta u_k \tag{6.9}$$

$$y_k = \overbrace{\begin{bmatrix}0 & 0 & 1\end{bmatrix}}^{\boldsymbol{C}_e}\boldsymbol{x}^e_k \tag{6.10}$$

Subsequently, an MPC problem can be formulated as:

$$J = \min_{\boldsymbol{\Delta U}} \quad \boldsymbol{\Delta\Xi}^\top\boldsymbol{Q}_\xi\boldsymbol{\Delta\Xi} + (\boldsymbol{Z}-\boldsymbol{Z}^{\text{ref}})^\top\boldsymbol{Q}_z(\boldsymbol{Z}-\boldsymbol{Z}^{\text{ref}}) + \boldsymbol{\Delta U}^\top\boldsymbol{R}\boldsymbol{\Delta U} \tag{6.11}$$

where

$$\boldsymbol{Z}^{\text{ref}} = \begin{bmatrix}z^{x\,\text{ref}}(k+1) & z^{x\,\text{ref}}(k+2) & \cdots & z^{x\,\text{ref}}(k+N_p)\end{bmatrix}^\top \tag{6.12}$$

$$\boldsymbol{\Delta U} = \begin{bmatrix}\Delta u(k) & \Delta u(k+1) & \cdots & \Delta u(k+N_p-1)\end{bmatrix}^\top \tag{6.13}$$

$$\boldsymbol{Z} = \begin{bmatrix}\boldsymbol{C}_e\boldsymbol{A}_e\\ \boldsymbol{C}_e\boldsymbol{A}_e^2\\ \vdots\\ \boldsymbol{C}_e\boldsymbol{A}_e^{Np}\end{bmatrix}\boldsymbol{x}^e_k + \begin{bmatrix}\boldsymbol{C}_e\boldsymbol{B}_e & 0 & \dots & 0\\ \boldsymbol{C}_e\boldsymbol{A}_e\boldsymbol{B}_e & \boldsymbol{C}_e\boldsymbol{B}_e & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{C}_e\boldsymbol{A}_e^{Np-1}\boldsymbol{B}_e & \dots & \dots & \boldsymbol{C}_e\boldsymbol{B}_e\end{bmatrix}\boldsymbol{\Delta U} \tag{6.14}$$

$$\boldsymbol{\Delta\Xi} = \begin{bmatrix} \boldsymbol{C}_\xi \boldsymbol{A}_e \\ \boldsymbol{C}_\xi \boldsymbol{A}_e^2 \\ \vdots \\ \boldsymbol{C}_\xi \boldsymbol{A}_e^{Np} \end{bmatrix} \boldsymbol{x}_k^e + \begin{bmatrix} \boldsymbol{C}_\xi \boldsymbol{B}_e & 0 & \dots & 0 \\ \boldsymbol{C}_\xi \boldsymbol{A}_e \boldsymbol{B}_e & \boldsymbol{C}_\xi \boldsymbol{B}_e & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{C}_\xi \boldsymbol{A}_e^{Np-1} \boldsymbol{B}_e & \dots & \dots & \boldsymbol{C}_\xi \boldsymbol{B}_e \end{bmatrix} \boldsymbol{\Delta U} \tag{6.15}$$

with N_p be the prediction horizon and $\boldsymbol{C}_\xi = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$ be a selection matrix. Consequently, (6.11) can take the standard quadratic form:

$$J = \min_{\boldsymbol{\Delta U}} \boldsymbol{\Delta U}^\top \boldsymbol{H} \boldsymbol{\Delta U} + \boldsymbol{f}^\top \boldsymbol{\Delta U} \tag{6.16}$$

where

$$\boldsymbol{H} = \boldsymbol{\Phi}_{\xi u}^\top \boldsymbol{Q}_\xi \boldsymbol{\Phi}_{\xi u} + \boldsymbol{\Phi}_{zu}^\top \boldsymbol{Q}_z \boldsymbol{\Phi}_{zu} + \boldsymbol{R} \tag{6.17}$$

$$\boldsymbol{f}^\top = 2 \left(\boldsymbol{x}_k^{e\top} \boldsymbol{\Phi}_\xi^\top \boldsymbol{Q}_\xi \boldsymbol{\Phi}_{\xi u} + \boldsymbol{x}_k^{e\top} \boldsymbol{\Phi}_z^\top \boldsymbol{Q}_z \boldsymbol{\Phi}_{zu} - \boldsymbol{Z}^{\mathrm{ref}\top} \boldsymbol{Q}_z \boldsymbol{\Phi}_{zu} \right) \tag{6.18}$$

Quasi-static stability constraints can be added by constraining the ZMP to lie withing the convex polygon of the feet.

$$\boldsymbol{Z}_{\min} \leq \boldsymbol{\Phi}_{zu} \boldsymbol{\Delta U} + \boldsymbol{\Phi}_z \boldsymbol{x}_k^e \leq \boldsymbol{Z}_{\max} \tag{6.19}$$

where $\boldsymbol{Z}_{\min}$, $\boldsymbol{Z}_{\max}$ indicate the minimal and maximal bounds within the prediction horizon N_p.

In such a way, a stable walking pattern is generated once per step. Nevertheless, a humanoid can be subject to disturbances that commonly occur during the gait. To this end, further stabilization is needed.

The desired CoM trajectory is stabilized in two stages. First, the desired ZMP is stabilized according to the DCM control in [139]:

$$z^{x\,*} = z^{x\,d} - \left(1 + \tfrac{K_p}{\omega}\right) \left(\xi^{x\,d} - \hat{\xi}^x\right) - \frac{K_i}{\omega} \int \left(\xi^{x\,d} - \hat{\xi}^x\right) dt \tag{6.20}$$

where K_p and K_i are proportional and integral gains, $z^{x\,d}$ and $\xi^{x\,d}$ is the desired ZMP and DCM obtained from the pattern generator, and $\hat{\xi}^x$ is the estimated DCM with SEROW.

Subsequently, an admittance controller [140] stabilizes the desired CoM acceleration with ZMP feedback:

$$\ddot{c}^{x\,*} = \ddot{c}^{x\,d} + K_c \left(z^{x\,\mathrm{FSR}} - z^{x\,*}\right) \tag{6.21}$$

with K_c be a proportional gain, $\ddot{c}^{x\,d}$ is the desired CoM acceleration and $z^{x\,\mathrm{FSR}}$ the mea-

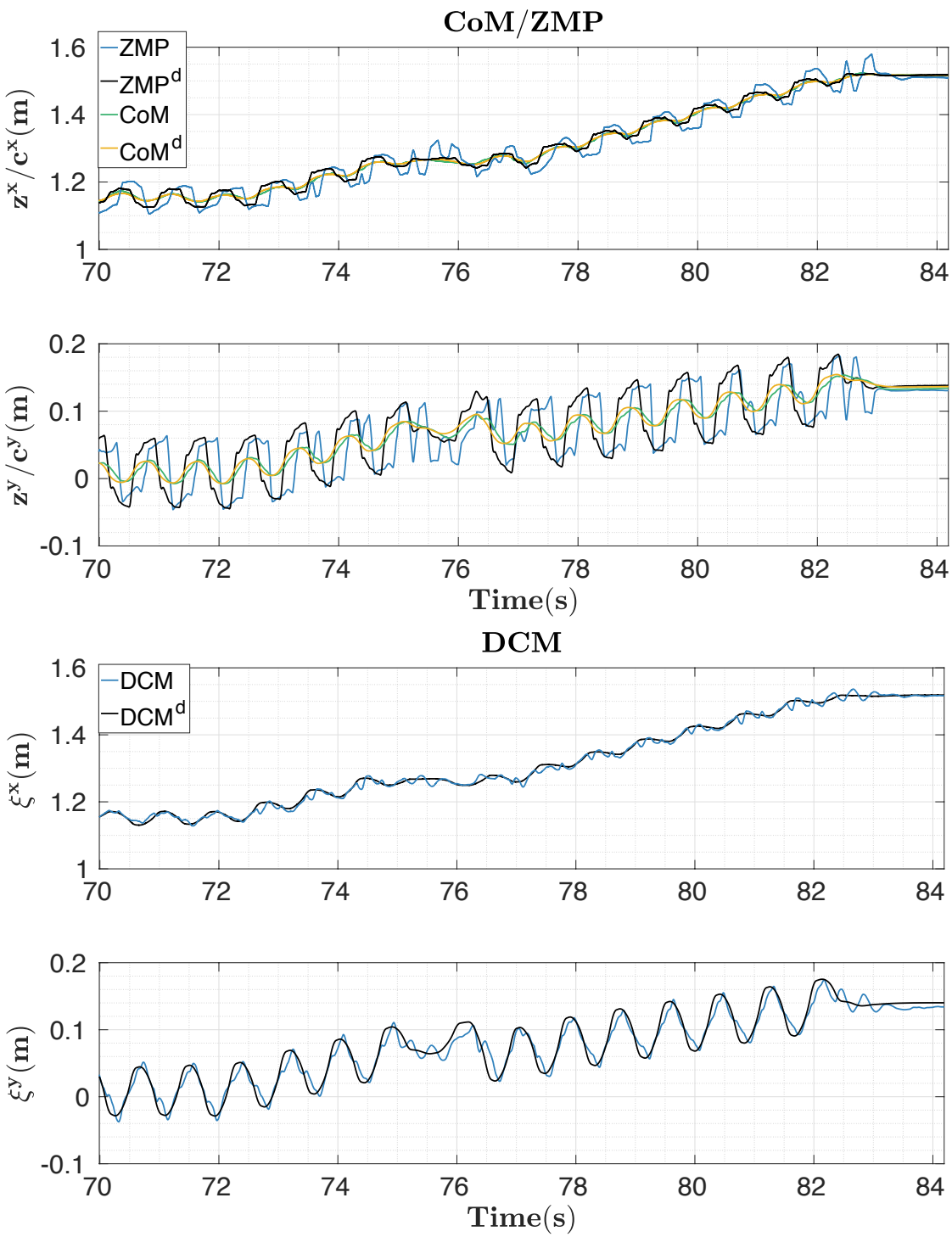

Figure 6.17: **Top:** 2D-CoM/ZMP: blue/black lines indicate the measured and desired ZMP, while green/orange lines are the estimated with SEROW and desired CoM trajectories. **Bottom:** 2D-DCM: blue/black are the estimated with SEROW and desired DCM trajectories respectively.

sured ZMP with the FSRs. The desired ZMP and CoM trajectories along with the measured ZMP and the estimated CoM are shown in Figure 6.17 top for a segment of the lab navigation experiment. Additionally, Figure 6.17 bottom illustrates the desired and estimated DCM trajectories.

Next, the desired joint states are computed with inverse kinematics from the target CoM and the target arm/feet poses. So far this control scheme facilitates walking over flat terrain. Nevertheless, when omni-directional over uneven terrain is considered, further postural stabilization is mandatory. Therefore, we regulate the body orientation with a damping control similarly to [141]:

$$\delta\dot{\phi}_b = K_{cx}\left(\phi_b^d - \phi_b\right) - \frac{1}{T_{cx}}\delta\phi_b \tag{6.22}$$

$$\delta\dot{\theta}_b = K_{cy}\left(\theta_b^d - \theta_b\right) - \frac{1}{T_{cy}}\delta\theta_b \tag{6.23}$$

where ϕ_b^d, θ_b^d and ϕ_b, θ_b are the desired and estimated roll and pitch angles respectively. K_{cx}, K_{cy} are proportional gains and T_{cx}, T_{cy} are time constants to reach neutral points.

In this work, we chose to directly regulate the joint space rather than the task space e.g. body/feet orientation as in [141]. In the joint space, the joint limits are known in advance and thus one can easily observe when the controllers saturate. In addition, inverse kinematics can be inaccurate, since implicitly it is assumed that the robot is fully rigid. The latter is not the case, every robot has a degree of compliance/elasticity.

To this end, for each leg in contact, the hip roll and pitch joints are modified as:

$$\mathrm{LHipRoll} = \mathrm{LHipRoll^d} - \delta\phi_\mathrm{b} \tag{6.24}$$

$$\mathrm{LHipPitch} = \mathrm{LHipPitch^d} - \delta\theta_\mathrm{b} \tag{6.25}$$

$$\mathrm{RHipRoll} = \mathrm{RHipRoll^d} - \delta\phi_\mathrm{b} \tag{6.26}$$

$$\mathrm{RHipPitch} = \mathrm{RHipPitch^d} - \delta\theta_\mathrm{b} \tag{6.27}$$

Subsequently, we employ the ZMP distributor, proposed in [141], to compute the desired ankle torque τ_l^d and τ_r^d for the left and right foot respectively. A similar damping control is used to realize the desired torques:

$$\delta\dot{\tau}_{ix} = K_{ax}\left(\tau_{ix}^d - \tau_{ix}\right) - \frac{1}{T_{ax}}\delta\tau_{ix} \tag{6.28}$$

$$\delta\dot{\tau}_{iy} = K_{ay}\left(\tau_{iy}^d - \tau_{iy}\right) - \frac{1}{T_{ay}}\delta\tau_{iy} \quad \text{where i} = \mathrm{l, r} \tag{6.29}$$

where as before K_{ax}, K_{ay} are proportional gains and T_{ax}, T_{ay} times constants to reach neutral points, while τ_l, τ_r are the measured left and right foot ankle torques.

Accordingly, for each leg in contact, the ankle roll and pitch joints are modified as:

$$\mathrm{LAnkleRoll} = \mathrm{LAnkleRoll}^{\mathrm{d}} - \delta\tau_{\mathrm{lx}} \tag{6.30}$$

$$\mathrm{LAnklePitch} = \mathrm{LAnklePitch}^{\mathrm{d}} - \delta\tau_{\mathrm{ly}} \tag{6.31}$$

$$\mathrm{RAnkleRoll} = \mathrm{RAnkleRoll}^{\mathrm{d}} - \delta\tau_{\mathrm{rx}} \tag{6.32}$$

$$\mathrm{RAnklePitch} = \mathrm{RAnklePitch}^{\mathrm{d}} - \delta\tau_{\mathrm{ry}} \tag{6.33}$$

Finally, the modified joint angles are tracked on NAO with a PID control loop at $100hz$.

Chapter 7
Conclusions

The art and science of asking questions is the source of all knowledge.
Thomas Berger (1924–2014)

7.1 Summary

In this thesis, a complete state estimation framework for humanoid robot walking over uneven terrain was presented. Initially, we considered the humanoid's centroidal dynamics and rigorously derived a nonlinear CoM estimator without relying on simplifying dynamic assumptions. The latter fuses joint encoder, IMU, pressure measurements to accurately estimate the 3D-CoM position, velocity, and external forces acting on the CoM, while directly considering the presence of uneven terrain and the CoM's angular momentum rate and, thus, effectively coupling the frontal with the lateral plane dynamics. To the best of our knowledge this is the first 3D-CoM estimation scheme that does not rely on F/T sensing at the robot's feet for the task at hand and at the present time constitutes one out of three 3D-CoM estimation schemes available worldwide.

Subsequently, we extended an established floating mass estimator to take into account the support foot pose, yielding in such a way the mandatory, for CoM estimation, affine transformations of the base and the support foot and forming a cascade state estimation scheme coined State Estimation RObot Walking (SEROW).

In order to robustify base state estimation against Visual Odometry/LIDAR Odometry outliers, we derived an analytical outlier detection algorithm in the context of the EKF. The latter approach termed as Robust Gaussian Error State Kalman Filter (RGESKF) proved to be robust for humanoid walking in dynamic environments where the 'static world assumption' speculated in most contemporary VO/LO algorithms does not hold. Since floating mass base estimation is fluently utilized both in UAVs and mobile/marine robotics, the RGESKF can be readily generalized to other robotic platforms beside humanoids.

Up to date control and state estimation schemes readily assume that feet contact status is known in a-priori. Nevertheless, contact detection is an important topic and in our opinion remains largely unexplored in the humanoid robotic literature. In this thesis, we elaborated for the first time on a broader question: *in which gait phase is the robot currently in?* To this end, we introduced an unsupervised learning framework called Gait-Phase Estimation Module (GEM) for gait phase estimation based solely on proprioceptive sensing, namely joint encoder, IMU and F/T data. Additionally, in our analysis we observed that a lower dimension latent space can appropriately represent the higher dimension gait phase dynamics. This observations along with similar remarks reported in [111] and the success of simplifying dynamic models, such as the LIPM [10], in walking, indicates that locomotion may be a low-dimensional skill.

7.2 Future Work

7.2.1 CoM Estimation

Since humanoid robot locomotion is hybrid and highly nonlinear by nature, possible future work aims at considering other nonlinear estimation techniques, such as the UKF or the particle filter, to further increase the estimation accuracy and also overcome the input-output correlation described in Chapter 3. Additionally, the Invariant EKF [142] and the Invariant UKF [143] have been employed in localization for mobile robots/UAVs and faster convergence properties have been demonstrated when contrasted to the EKF/UKF. Recently, such approaches proved effective also in humanoid base estimation [144, 145], thus it is worth investigating whether geometrical properties such as left or right invariance also apply in CoM estimation. Finally, it would be interesting to employ gyros in every link and determine how much the estimation accuracy increases when incorporating a direct measurement for the CoM velocity.

7.2.2 Base Estimation

Planned future work regards, a) studying how base drift accumulates when the humanoid walks over terrains with various unevenness and/or compliance, b) utilizing the proposed scheme in humanoid navigation in human/gradually changing environments, where outliers are harder to detect [77]. In addition, the study of estimation performance in locomotion manipulation tasks, where the humanoid has to walk to a desired location and grasp an object is interesting. Commonly while grasping, multiple self-parts appear in the FoV and possibly give rise to outliers. Furthermore, how to recover from VO/LO divergence and continue integrating VO/LO measurements needs to also be addressed. Finally, it is promising to investigate whether mounting additional IMUs on the feet and properly consider them in the dynamical model does increase base estimation accuracy when slippage

on the ground occurs.

7.2.3 Gait Phase Estimation

Regarding gait phase estimation, future work aims at identifying a) the latent space components with a factor analysis and b) the latent gait phase dynamics by fitting gaussian processes, to directly utilize them in our control and state estimation scheme. In addition, how the change of friction coefficients influences the classification accuracy needs to be investigated. Moreover, one should consider how the proposed gait phase estimation scheme can be transferred to platforms that are not employed with 6D F/T in the feet and thus the full contact wrench cannot be directly measured. Subsequently, it is important to examine how many latent dimensions are sufficient for accurately estimating the gait phase while dynamic walking. Finally, it worths investigating whether considering the feet linear accelerations/angular velocities in the estimation procedure increases classification accuracy during dynamic walking gaits.

7.2.4 Emerging Topics

Through our research many interesting topics naturally emerged. More specifically, we've identified that the proposed humanoid state estimation scheme, can be transfered to other legged robotic platforms such as quadrupeds, hexapods etc. Both the CoM and gait phase estimation originate from the nonlinear 6D centroidal Newton-Euler dynamics, while the base estimation is formulated from the nonlinear dynamics of a floating mass. The latter are valid for legged robots and are indenpended of the number of legs. To this end, the proposed state estimation scheme can be extended to legged robots that have at least an IMU in the base, joint encoders, and feet pressure or feet F/T sensors by properly considering the number of legs/contacts in the computations. We strongly believe in the future it will be important to have a single state estimator that accurately and efficiently addresses legged locomotion and is independed of the robot's morphology.

Additionally, we recognized that in state estimation without exterioceptive sensing, kinematic drift gives rise to inaccuracies. Commonly, this happens when the robot impacts the ground and generates enormous GRFs that in turn cause the feet to slip. Therefore, it would be interesting to adapt online the control strategy to facilitate more accurate state estimation. So far state estimation is considered as a mean to allow for feedback control and not the other way around. Thus, when the state estimator exhibits low confidence on the provided feedback it should be regarded as a sign of adapting the control in order to actively improve the state estimation accuracy, i.e. perform slower or smoother motions.

Finally, we formulated and implemented a robust and accurate nonlinear state estimation scheme for the case of a single robot. Morever, we've demonstrated its efficacy in individual modules such as VSLAM, footstep planning, and feedback control. Never-

theless, It would be extremely interesting to consider a multi-agent complex environment where robots have to collaborate in order to jointly accomplish a multi-tasking scenario. In the latter, multi-agent state estimation realizes a vital role in the task's outcome. Thus, for example, if a robot visually tracked the location and morphology [146] of the nearest one, it could deliver a relative base and CoM measurement. These measurements could be fused by the other robot and potentially improve its state estimation by correcting the drift.

Bibliography

[1] M. Vukobratovic and B. Borovac, "Zero-Moment Point – Thirty Five Years of its Life," *International Journal of Humanoid Robotics*, vol. 1, no. 01, pp. 157–173, 2004.

[2] A. Goswami, "Postural Stability of Biped Robots and the Foot Rotation Indicator (FRI) Point," *Robotics Research*, vol. 18, pp. 523–533, 1999.

[3] H. Hirukawa, S. Hattori, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, and M. Morisawa, "A Universal Stability Criterion of the Foot Contact of Legged Robots - Adios ZMP," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1976–1983, 2006.

[4] T. Koolen, T. De Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based Analysis and Control of Legged Locomotion, Part 1: Theory and Application to Three Simple Gait Models," *Int. J. Rob. Res.*, vol. 31, pp. 1094–1113, Aug. 2012.

[5] J.-P. Aubin, *Viability Theory*. Cambridge, MA, USA: Birkhauser Boston Inc., 1991.

[6] P.-B. Wieber, "Viability and Predictive Control for Safe Locomotion," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 1103–1108, Sept 2008.

[7] D. Hobbelen and M. Wisse, "A Disturbance Rejection Measure for Limit Cycle Walkers: The Gait Sensitivity Norm," *Robotics, IEEE Transactions on*, vol. 23, pp. 1213–1224, Dec 2007.

[8] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D Linear Inverted Pendulum Mode: A Simple Modeling for a Biped Walking Pattern Generation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, 2001.

[9] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa, "A realtime pattern generator for biped walking," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, pp. 31–37, 2002.

[10] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pp. 1620–1626, 2003.

[11] J. Strom, G. Slavov, and E. Chown, "Omnidirectional walking using ZMP and preview control for the NAO humanoid robot," in *RoboCup 2009: Robot Soccer World Cup XIII*, vol. 5949 of *Lecture Notes in Computer Science*, pp. 378–389, Springer, 2010.

[12] S. Czarnetzki, S. Kerner, and O. Urbann, "Observer-based Dynamic Walking Control for Biped Robots," *Robotics and Autonomous Systems*, vol. 57, no. 8, pp. 839–845, 2009.

[13] D. Dimitrov, P. B. Wieber, O. Stasse, H. Ferreau, and H. Diedam, "An optimized Linear Model Predictive Control solver for online walking motion generation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1171–1176, 2009.

[14] D. Dimitrov, A. Sherikov, and P.-B. Wieber, "A Sparse Model Predictive Control Formulation for Walking Motion Generation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2292–2299, 2011.

[15] S. Piperakis, E. Orfanoudakis, and M. Lagoudakis, "Predictive Control for Dynamic Locomotion of Real Humanoid Robots," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 4036–4043, Sept 2014.

[16] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," in *Proceedings of the 6th IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pp. 200–207, Dec 2006.

[17] A. L. Hof, "The 'extrapolated center of mass' concept suggests a simple control of balance in walking," *Human Movement Science*, vol. 27, pp. 112 – 125, 2008.

[18] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot -1st report: Walking gait pattern generation-," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pp. 1084–1091, 2009.

[19] J. Englsberger, C. Ott, M. Roa, A. Albu-Schaffer, and G. Hirzinger, "Bipedal Walking Control based on Capture Point dynamics," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4420–4427, Sept 2011.

[20] J. Englsberger, C. Ott, and A. Albu-Schaffer, "Three-dimensional bipedal walking control using divergent component of motion," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2600–2607, Nov 2013.

[21] T. Koolen, T. de Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple

gait models," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 2012.

[22] J. Pratt, T. Koolen, T. de Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus, "Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1117–1133, 2012.

[23] S. Caron and A. Kheddar, "Dynamic Walking over Rough Terrains by Nonlinear Predictive Control of the Floating-base Inverted Pendulum," *CoRR*, 2017.

[24] S. Caron, A. Escande, L. Lanari, and B. Mallein, "Capturability-based analysis, optimization and control of 3d bipedal walking," in *CoRR*, pp. 3293–3298, Jan 2018.

[25] S. Caron and B. Mallein, "Balance control using both zmp and com height variations: A convex boundedness approach," in *IEEE International Conference on Robotics and Automation*, pp. 3293–3298, May 2018.

[26] E. R. Westervelt, J. W. Grizzle, and C. Chevallereau, "Feedback control of dynamic bipedal robot locomotion. taylor and francis/crc," 2007.

[27] E. R. Westervelt, J. Grizzle, and D. Koditschek, "Hybrid Zero Dynamics of Planar Biped Walkers," *Automatic Control, IEEE Transactions on*, vol. 48, pp. 42–56, Jan 2003.

[28] C. Chevallereau, J. W. Grizzle, and C. H. Moog, "Nonlinear control of mechanical systems with one degree of underactuation," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 3, pp. 2222–2228 Vol.3, April 2004.

[29] B. G. Buss, A. Ramezani, K. A. Hamed, B. A. Griffin, K. S. Galloway, and J. W. Grizzle, "Preliminary walking experiments with underactuated 3d bipedal robot marlo," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2529–2536, Sept 2014.

[30] K. A. Hamed, B. G. Buss, and J. W. Grizzle, "Continuous-time controllers for stabilizing periodic orbits of hybrid systems: Application to an underactuated 3d bipedal robot," in *53rd IEEE Conference on Decision and Control*, pp. 1507–1513, Dec 2014.

[31] X. Da, O. Harib, R. Hartley, B. Griffin, and J. W. Grizzle, "From 2d design of underactuated bipedal gaits to 3d implementation: Walking with speed tracking," *IEEE Access*, vol. 4, pp. 3469–3478, 2016.

[32] R. Hartley, X. Da, and J. W. Grizzle, "Stabilization of 3d underactuated biped robots: Using posture adjustment and gait libraries to reject velocity disturbances," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 1262–1269, Aug 2017.

[33] N. Sadati, G. A. Dumont, K. A. Hamed, and W. A. Gruver, *Hybrid Control and Motion Planning of Dynamical Legged Locomotion*. Piscataway, NJ, USA: Wiley IEEE Press Series on Systems Science and Engineering, 2012.

[34] J. Reher, E. A. Cousineau, A. Hereid, C. M. Hubicki, and A. D. Ames, "Realizing dynamic and efficient bipedal locomotion on the humanoid robot DURUS," in *2016 IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pp. 1794–1801, May 2016.

[35] A. Hereid, E. Cousineau, C. Hubicki, and A. D. Ames, "3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 1447–1454, IEEE, 2016.

[36] A. Hereid, C. Hubicki, E. Cousineau, and A. D. Ames, "Dynamic Humanoid Locomotion: A Scalable Formulation for HZD Gait Optimization," *IEEE Transactions on Robotics*, 2018.

[37] L. Pongsak, M. Okada, and Y. Nakamura, "Optimal Filtering for Humanoid Robot State Estimators," 2002.

[38] B. J. Stephens, "State Estimation for Force-controlled Humanoid Balance using Simple Models in the Presence of Modeling Error," in *IEEE Intl. Conf. on Robotics and Automation*, pp. 3994–3999, 2011.

[39] S. Kajita and K. Tani, "Study of Dynamic Biped Locomotion on Rugged Terrain-derivation and Application of the Linear Inverted Pendulum Mode," in *IEEE Intl. Conf. on Robotics and Automation*, pp. 1405–1411, 1991.

[40] X. Xinjilefu and C. G. Atkeson, "State Estimation of a Walking Humanoid Robot," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pp. 3693–3699, 2012.

[41] S. Kwon and Y. Oh, "Estimation of the Center of Mass of Humanoid Robot," in *International Conference on Control, Automation and Systems*, pp. 2705–2709, 2007.

[42] R. Wittmann, A. Hildebrandt, D. Wahrmann, D. Rixen, and T. Buschmann, "State Estimation for Biped Robots using Multibody Dynamics," in *IEEE/RSJ Conf. Intel. Robots and Systems*, 2015.

[43] H. Bae and J.-H. Oh, "Biped robot state estimation using compliant inverted pendulum model," *Robotics and Autonomous Systems*, vol. 108, pp. 38 – 50, 2018.

[44] H. Bae, H. Jeong, J. Oh, K. Lee, and J. Oh, "Humanoid robot com kinematics estimation based on compliant inverted pendulum model and robust state estimator," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 747–753, Oct 2018.

[45] J. Carpintieri, M. Benallegue, N. Mansard, and J. P. Laumond, "Center-of-Mass Estimation for a Polyarticulated System in Contact - A Spectral Approach," *IEEE Transactions on Robotics*, vol. 32, 2016.

[46] N. Rotella, A. Herzog, S. Schaal, and L. Righetti, "Humanoid momentum estimation using sensed contact wrenches," in *IEEE-RAS Intl. Conf. on Humanoid Robots*, pp. 556–63, 2015.

[47] S. Piperakis and P. Trahanias, "Non-linear ZMP based State Estimation for Humanoid Robot Locomotion," in *IEEE-RAS Intl. Conf. on Humanoid Robots*, pp. 202–209, 2016.

[48] S. Piperakis and P. Trahanias, "Cascade Non-Linear State Estimation for Humanoid Robot Locomotion," *Proceedings of the Dynamic Walking*, June 2017.

[49] S. Piperakis, M. Koskinopoulou, and P. Trahanias, "Nonlinear State Estimation for Humanoid Robot Walking," *IEEE Robotics and Automation Letters*, vol. 3, pp. 3347–3354, Oct 2018.

[50] Y. Aoustin, F. Plestan, and V. Lebastard, "Experimental comparison of several posture estimation solutions for biped robot Rabbit," in *2008 IEEE International Conference on Robotics and Automation*, pp. 1270–1275, 2008.

[51] R. Mahony, T. Hamel, and J. Pflimlin, "Nonlinear Complementary Filters on the Special Orthogonal Group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[52] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*, pp. 1–7, 2011.

[53] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based Locomotion Planning, Estimation, and Control Design for the Atlas Humanoid Robot," *Autonomous Robots*, vol. 40, pp. 429–455, 2016.

[54] M. F. Fallon, M. Antone, N. Roy, and S. Teller, "Drift-free Humanoid State Estimation Fusing Kinematic, Inertial and LIDAR Sensing," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 112–119, 2014.

[55] M. F. Fallon, P. Marion, R. Deits, T. Whelan, M. Antone, J. McDonald, and R. Tedrake, "Continuous Humanoid Locomotion over Uneven Terrain using Stereo Fusion," in *IEEE-RAS Intl. Conf. on Humanoid Robots*, pp. 881–888, 2015.

[56] M. Bloesch, M. Hutter, M. Hoepflinger, S. Leutenegger, C. Gehring, D. Remy, and R. Siegwart, "State Estimation for Legged Robots–Consistent Fusion of Leg Kinematics and IMU ," *Robotics Sci. and Sys.*, 2012.

[57] N. Rotella, M. Bloesch, L. Righetti, and S. Schaal, "State Estimation for a Humanoid Robot," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pp. 952–958, 2014.

[58] M. Benallegue and F. Lamiraux, "Humanoid Flexibility Deformation can be efficiently estimated using only Inertial Measurement Units and Contact Information," in *2014 IEEE-RAS International Conference on Humanoid Robots*, pp. 246–251, 2014.

[59] X. Xinjilefu, S. Feng, and C. G. Atkeson, "Dynamic State Estimation using Quadratic Programming," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems,*, pp. 989–994, 2014.

[60] S. Piperakis, D. Kanoulas, N. G. Tsagarakis, and P. Trahanias, "Outlier-Robust State Estimation for Humanoid Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019.

[61] M. Neunert, M. Stauble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, pp. 1458–1465, July 2018.

[62] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Auton. Robots*, vol. 40, pp. 473–491, Mar. 2016.

[63] T. Koolen, S. Bertrand, G. Thomas, T. de Boer, T. Wu, J. Smith, J. Englsberger, and J. Pratt, "Design of a Momentum-Based Control Framework and Application to the Humanoid Robot Atlas," *Intl. Journal of Humanoid Robotics*, vol. 13, p. 1650007, 2016.

[64] B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernandez-Lopez, and C. Semini, "Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer

convex optimization," *IEEE Robotics and Automation Letters*, vol. 3, pp. 2531–2538, July 2018.

[65] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, pp. 1560–1567, July 2018.

[66] P.-B. Wieber, "Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pp. 137–142, Dec 2006.

[67] S. Piperakis, E. Orfanoudakis, and M. Lagoudakis, "Predictive Control for Dynamic Locomotion of Real Humanoid Robots," in *IEEE/RSJ Intl. Conf. Intel. Robots and Systems*, pp. 4036–4043, 2014.

[68] A. Bowling, "Impact forces and mobility in legged robot locomotion," in *2007 IEEE/ASME international conference on advanced intelligent mechatronics*, pp. 1–8, Sept 2007.

[69] A. Petrovskaya, J. Park, and O. Khatib, "Probabilistic estimation of whole body contacts for multi-contact robot control," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 568–573, April 2007.

[70] V. Ortenzi, H. Lin, M. Azad, R. Stolkin, J. A. Kuo, and M. Mistry, "Kinematics-based estimation of contact constraints using only proprioception," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 1304–1311, Nov 2016.

[71] J. Hwangbo, C. D. Bellicoso, P. Fankhauser, and M. Huttery, "Probabilistic foot contact estimation by fusing information from dynamics and differential/forward kinematics," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3872–3878, Oct 2016.

[72] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, "Trajectory optimization through contacts and automatic gait discovery for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 2, pp. 1502–1509, July 2017.

[73] M. Camurri, M. Fallon, S. Bazeille, A. Radulescu, V. Barasuol, D. G. Caldwell, and C. Semini, "Probabilistic contact estimation and impact detection for state estimation of quadruped robots," *IEEE Robotics and Automation Letters*, vol. 2, pp. 1023–1030, April 2017.

[74] M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger, and R. Siegwart, "State Estimation for Legged Robots on Unstable and Slippery Terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 6058–6064, 2013.

[75] N. Rotella, S. Schaal, and L. Righetti, "Unsupervised Contact Learning for Humanoid Estimation and Control," in *Proceedings of the IEEE Intl. Conf. on Robotics and Automation*, 2018.

[76] S. Piperakis, S. Timotheatos, and P. Trahanias, "Unsupervised Gait Phase Estimation for Humanoid Robot Walking," in *IEEE Intl. Conf. on Robotics and Automation*, 2019.

[77] S. Piperakis, N. Tavoularis, E. Hourdakis, D. Kanoulas, and P. Trahanias, "Humanoid Robot Dense RGB-D SLAM for Embedded Devices," in *IEEE Intl. Conf. on Robotics and Automation (ICRA): 3rd Full-Day Workshop "Towards Real-World Deployment of Legged Robots"*, 2019.

[78] S. Timotheatos, S. Piperakis, A. Argyros, and P. Trahanias, "Vision Based Horizon Detection for UAV Navigation," in *Advances in Service and Industrial Robotics*, (Cham), pp. 181–189, Springer International Publishing, 2019.

[79] S. Timotheatos , S. Piperakis, and P. Trahanias, "Visual horizon line detection for uav navigation," *International Journal of Mechanics and Control*, vol. 20, pp. 35–51, June 2019.

[80] M. Koskinopoulou, S. Piperakis, and P. Trahanias, "Learning from Demonstration facilitates Human-Robot Collaborative task execution," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 59–66, March 2016.

[81] M. Hopkins, D. Hong, and A. Leonessa, "Humanoid locomotion on uneven terrain using the time-varying divergent component of motion," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pp. 266–272, Nov 2014.

[82] Z. Chen, K. Jiang, and J. C. Hung, "Local Observability Matrix and its Application to Observability Analyses," in *16th Annual Conference of IEEE Industrial Electronics Society*, pp. 100–103 vol.1, 1990.

[83] T. Niemüller, A. Ferrein, G. Eckel, D. Pirro, P. Podbregar, T. Kellner, C. Rath, and G. Steinbauer, "Providing Ground-truth Data for the Nao Robot Platform," in *RoboCup 2010* (J. Ruiz-del Solar, E. Chown, and P. G. Plöger, eds.), pp. 133–144, Springer-Verlag, 2011.

[84] N. El-Sheimy, H. Hou, and X. Niu, "Analysis and Modeling of Inertial Sensors Using Allan Variance," *IEEE Transactions on Instrumentation and Measurement*, pp. 140–149, 2008.

[85] R. Martinez-Cantin and J. A. Castellanos, "Bounding Uncertainty in EKF-SLAM: the Robocentric Local Approach," in *Intl. Conf. on Robotics and Automation*, pp. 430–435, 2006.

[86] X. Xinjilefu, S. Feng, and C. G. Atkeson, "Center of Mass Estimator for Humanoids and its Application in Modelling Error Compensation, Fall Detection and Prevention," in *IEEE-RAS Intel. Conf. on Humanoid Robots*, pp. 67–73, 2015.

[87] S. Kajita et al., "Biped Walking Pattern Generation based on Spatially Quantized Dynamics," in *IEEE-RAS Intl. Conf. on Humanoid Robots*, pp. 599–605, 2017.

[88] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[89] J. Englsberger, T. Koolen, S. Bertrand, J. Pratt, C. Ott, and A. Albu-Schaffer, "Trajectory generation for continuous leg forces during double support and heel-to-toe shift based on divergent component of motion," in *IEEE/RSJ Conf. Intel. Robots and Systems*, 2014.

[90] S. Piperakis, "SEROW: State Estimation RObot Walking." https://github.com/mrsp/serow.

[91] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast Semi-direct Monocular Visual Odometry," in *IEEE Intl. Conf. on Robotics and Automation*, pp. 15–22, 2014.

[92] J. Engel, V. Koltun, and D. Cremers, "Direct Sparse Odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 611–625, 2018.

[93] M. Jaimez, J. G. Monroy, and J. Gonzalez-Jimenez, "Planar Odometry from a Radial Laser Scanner. A Range Flow-based Approach," in *IEEE Intl. Conf. on Robotics and Automation*, pp. 4479–4485, 2016.

[94] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robotics: Science and Systems*, 2014.

[95] R. Piche, S. Sarkka, and J. Hartikainen, "Recursive Outlier-robust Filtering and Smoothing for Nonlinear Systems using the Multivariate Student-t Distribution," in *IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6, 2012.

[96] H. Wang, H. Li, J. Fang, and H. Wang, "Robust Gaussian Kalman Filter With Outlier Detection," *IEEE Signal Processing Letters*, vol. 25, no. 8, pp. 1236–1240, 2018.

[97] N. G. Tsagarakis *et al.*, "WALK-MAN: A High-Performance Humanoid Platform for Realistic Environments," *Journal of Field Robotics*, vol. 34, no. 7, pp. 1225–1259, 2017.

[98] X. Ding, L. He, and L. Carin, "Bayesian Robust Principal Component Analysis," in *IEEE Trans. Image Process*, vol. 20, pp. 3419–3430, 2011.

[99] Q. Wan, H. Duan, J. Fang, H. Li, and Z. Xing, "Robust Bayesian Compressed Sensing with Outliers," in *Signal Processing*, vol. 140, pp. 104–109, 2017.

[100] J. Paisley and L. Carin, "Nonparametric Factor Analysis with Beta Process Priors," in *26th Annual International Conference on Machine Learning*, (New York, NY, USA), pp. 777–784, ACM, 2009.

[101] I. Arasaratnam and S. Haykin, "Cubature Kalman Filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[102] C. Zhou, X. Wang, Z. Li, and N. Tsagarakis, "Overview of GaitSynthesis for the Humanoid COMAN," vol. 14, pp. 15–25, 2017.

[103] D. Kanoulas, C. Zhou, A. Nguyen, G. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Vision-based Foothold Contact Reasoning using Curved Surface Patches," in *IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pp. 121–128, 2017.

[104] L. Muratore, A. Laurenzi, E. Hoffman, A. Rocchi, D. G. Caldwell, and N. G. Tsagarakis, "Xbotcore: A Real-Time Cross-Robot Software Platform," in *IEEE International Conference on Robotic Computing (IRC)*, pp. 77–80, 2017.

[105] A. Rocchi, E. M. Hoffman, D. G. Caldwell, and N. G. Tsagarakis, "OpenSoT: A Whole-body Control Library for the Compliant Humanoid Robot COMAN," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6248–6253, 2015.

[106] N. El-Sheimy, H. Hou, and X. Niu, "Analysis and Modeling of Inertial Sensors Using Allan Variance," *IEEE Transactions on Instrumentation and Measurement*, pp. 140–149, 2008.

[107] A. Bry, A. Bachrach, and N. Roy, "State estimation for aggressive flight in gps-denied environments using onboard sensing," in *IEEE International Conference on Robotics and Automation*, pp. 1–8, 2012.

[108] S. Piperakis, "GEM: Gait-phase Estimation Module." https://github.com/mrsp/gem.

[109] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[110] D. DeMers and G. Cottrell, "Non-linear Dimensionality Reduction," in *Proceedings of the Intl. Conf. on Neural Information Processing Systems*, pp. 580–587, 1992.

[111] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, "Hybrid Zero Dynamics of Planar Biped Walkers," *IEEE Transactions on Automatic Control*, vol. 48, pp. 42–56, Jan 2003.

[112] W. Gouda, W. Gomaa, and T. Ogawa, "Vision based SLAM for Humanoid Robots: A Survey," in *IEEE JEC-ECC*, pp. 170–175, 2013.

[113] S. Kagami, K. Nishiwaki, J. Kuffner, K. Okada, M. Inaba, and H. Inoue, "Vision-based 2.5D Terrain Modeling for Humanoid Locomotion," in *IEEE ICRA*, pp. 2141–2146, 2003.

[114] O. Stasse, F. Saidi, K. Yokoi, B. Verrelst, B. Vanderborght, A. Davison, N. Mansard, and C. Esteves, "Integrating Walking and Vision to Increase Humanoid Autonomy," *IJHR*, vol. 5, no. 02, pp. 287–310, 2008.

[115] E. Wirbel, B. Steux, S. Bonnabel, and A. de La Fortelle, "Humanoid robot navigation: From a visual SLAM to a visual compass," in *2013 10th IEEE INTERNATIONAL CONFERENCE ON NETWORKING, SENSING AND CONTROL (ICNSC)*, pp. 678–683, 2013.

[116] D. Maier, C. Lutz, and M. Bennewitz, "Integrated Perception, Mapping, and Footstep Planning for Humanoid Navigation among 3D Obstacles," in *IEEE/RSJ IROS*, pp. 2658–2664, 2013.

[117] G. Oriolo, A. Paolillo, L. Rosa, and M. Vendittelli, "Vision-based Odometric Localization for Humanoids using a Kinematic EKF," in *12th IEEE-RAS ICHR*, pp. 153–158, 2012.

[118] T. Zhang, E. Uchiyama, and Y. Nakamura, "Dense RGB-D SLAM for Humanoid Robots in the Dynamic Humans Environment," in *IEEE-RAS 18th ICHR*, pp. 270–276, 2018.

[119] R. Scona, S. Nobili, Y. R. Petillot, and M. Fallon, "Direct Visual SLAM Fusing Proprioception for a Humanoid Robot," in *IEEE/RSJ IROS*, pp. 1419–1426, 2017.

[120] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera," in *24th Annual ACM Symposium on UIST*, pp. 559–568, 2011.

[121] D. Kanoulas, N. G. Tsagarakis, and M. Vona, "rxKinFu: Moving Volume KinectFusion for 3D Perception and Robotics," in *IEEE-RAS 18th ICHR*, pp. 1002–1009, 2018.

[122] N. Tsagarakis, D. Caldwell, A. Bicchi, F. Negrello, M. Garabini, W. Choi, L. Baccelliere, G. Vo, J. Noorden, M. Catalano, M. Ferrati, L. Muratore, A. Margan, L. Natale, E. Mingo, H. Dallali, A. Settimi, A. Rocchi, V. Varricchio, and D. Kanoulas, "Walk-man: A high performance humanoid platform for realistic environments," *Journal of Field Robotics*, 06 2017.

[123] N. Tsagarakis, F. Negrello, M. Garabini, W. Choi, L. Baccelliere, G. Vo, J. Noorden, M. Catalano, M. Ferrati, L. Muratore, P. Kryczka, E. Hoffman, A. Settimi, A. Rocchi, A. Margan, S. Cordasco, D. Kanoulas, A. Cardellino, L. Natale, and A. Bicchi, *WALK-MAN humanoid platform*, pp. 495–548. 04 2018.

[124] F. Negrello, A. Settimi, D. Caporale, G. Lentini, M. Poggiani, D. Kanoulas, L. Muratore, E. Luberto, G. Santaera, L. Ciarleglio, L. Ermini, L. Pallottino, D. G. Caldwell, N. Tsagarakis, A. Bicchi, M. Garabini, and M. G. Catalano, "Humanoids at Work: The WALK-MAN Robot in a Postearthquake Scenario," *IEEE Robotics Automation Magazine*, vol. 25, pp. 8–22, Sep. 2018.

[125] K. Walas, D. Kanoulas, and P. Kryczka, "Terrain classification and locomotion parameters adaptation for humanoid robots using force/torque sensing," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 133–140, Nov 2016.

[126] D. Kanoulas, A. Stumpf, V. Raghavan, C. Zhou, A. Toumpa, O. Stryk, D. Caldwell, and N. Tsagarakis, "Footstep planning in rough terrain for bipedal robots using curved contact patches," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–9, 2018.

[127] D. Kanoulas, C. Zhou, A. Nguyen, G. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Vision-based foothold contact reasoning using curved surface patches," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pp. 121–128, Nov 2017.

[128] D. Gouaillier and P. Blazevic, "A mechatronic platform, the Aldebaran Robotics humanoid robot," in *Proceedings of the 32nd IEEE Annual Conference on Industrial Electronics (IECON)*, pp. 4049–4053, 2006.

[129] Aldebaran Robotics, "Nao documentation," 2012. Only available online: www.aldebaran-robotics.com/documentation.

[130] Greek Robocup SPL team Kouretes, 2006. http://www.intelligence.tuc.gr/kouretes/web/.

[131] S. Piperakis, "nao_walk: 3D omni-directional walk engine for NAO humanoid robot." https://github.com/mrsp/nao_walk.

[132] K. Robotics, "JACO Assistive robotic arm." https://www.kinovarobotics.com.

[133] J. Garimort, A. Hornung, and M. Bennewitz, "Humanoid Navigation with Dynamic Footstep Plans," pp. 3982–3987, 05 2011.

[134] A. Hornung, A. Dornbush, M. Likhachev, and M. Bennewitz, "Anytime Search-based Footstep Planning with Suboptimality Bounds," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pp. 674–679, Nov 2012.

[135] A. Stumpf, S. Kohlbrecher, D. Conner, and O. Stryk, "Open Source Integrated 3D Footstep Planning Framework for Humanoid Robots," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*, pp. 938–945, IEEE, 2016.

[136] B. P. Gerkey, "ROS OpenSlam Gmapping." http://wiki.ros.org/gmapping.

[137] B. P. Gerkey, "ROS Adaptive Monte Carlo localization." http://wiki.ros.org/amcl.

[138] M. Khadiv, A. Herzog, S. A. A. Moosavian, and L. Righetti, "Step Timing Adjustment: A Step Toward Generating Robust Gaits," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 35–42, Nov 2016.

[139] J. Englsberger, T. Koolen, S. Bertrand, J. Pratt, C. Ott, and A. Albu-Schaffer, "Trajectory generation for continuous leg forces during double support and heel-to-toe shift based on divergent component of motion," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 4022–4029, Sept 2014.

[140] S. Caron, A. Kheddar, and O. Tempier, "Stair Climbing Stabilization of the HRP-4 Humanoid Robot using Whole-body Admittance Control," in *IEEE International Conference on Robotics and Automation*, 2019.

[141] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi, "Biped walking stabilization based on linear inverted pendulum tracking," pp. 4489 – 4496, 2010.

[142] A. Barrau and S. Bonnabel, "The invariant extended kalman filter as a stable observer," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, 2017.

[143] J. Condomines, C. Seren, and G. Hattenberger, "Invariant unscented Kalman filter with application to attitude estimation," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 2783–2788, 2017.

[144] R. Hartley, M. Jadidi, J. Grizzle, and R. Eustice, "Contact-aided invariant extended kalman filtering for legged robot state estimation," in *Proceedings of Robotics: Science and Systems*, (Pittsburgh, Pennsylvania), June 2018.

[145] R. Hartley, M. G. Jadidi, R. M. Eustice, and J. W. Grizzle, "Contact-Aided Invariant Extended Kalman Filtering for Robot State Estimation," *CoRR*, 2019.

[146] A. Qammaz and A. A. Argyros, "MocapNET: Ensemble of SNN Encoders for 3D Human Pose Estimation in RGB Images," in *British Machine Vision Conference (BMVC 2019)*, BMVA, September 2019.

[147] MemmoEU, "Memory of Motion (memmo)." www.memmo-project.eul.

[148] S. Caron, "Wrench Friction Cones." https://scaron.info/teaching/wrench-friction-cones.html.

[149] S. Caron, Q. Pham, and Y. Nakamura, "ZMP Support Areas for Multi-contact Mobility Under Frictional Constraints," *IEEE Transactions on Robotics*, vol. 33, pp. 67–80, Feb. 2017.

[150] S. Caron and A. Kheddar, "Multi-contact walking pattern generation based on model preview control of 3d com accelerations," in *IEEE-RAS International Conference on Humanoid Robots*, Nov. 2016.

[151] J. Pratt, T. Koolen, T. De Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus, "Capturability-based Analysis and Control of Legged Locomotion, Part 2: Application to M2V2, a Lower-body Humanoid," *Int. J. Rob. Res.*, vol. 31, pp. 1117–1133, Sept. 2012.

[152] J.-P. Aubin, J. Lygeros, M. Quincampoix, S. Sastry, and N. Seube, "Impulse differential inclusions: a viability approach to hybrid systems," *Automatic Control, IEEE Transactions on*, vol. 47, pp. 2–20, Jan 2002.

[153] M. Vukobratovic, *Biped Locomotion.* Springer-Verlag New York, Inc., 1990.

[154] T. McGeer, "Passive walking with knees," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, pp. 1640–1645, 1990.

[155] A. Schwab and M. Wisse, "Basin of Attraction of the Simplest Walking Model," *Proceedings of the ASME Design Engineering Technical Conference*, vol. 6, pp. 531–539, 2001.

[156] J. M. Ottino and T. Shinbrot, "Nonlinear dynamics and chaos (with applications to physics, biology chemistry, and engineering)," *AIChE Journal*, vol. 41, no. 7, pp. 1831–1832, 1995.

[157] K. Masuya and T. Sugihara, "A Dual-Stage Complementary Filter for Dead Reckoning of a Biped Robot via Estimated Contact Point," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 112–117, 2013.

[158] K. Masuya and T. Sugihara, "Dead Reckoning for Biped Robots that Suffers Less from Foot Contact Condition based on Anchoring Pivot Estimation," *Advanced Robotics*, vol. 29, pp. 1–15, 03 2015.

[159] R. Hermann and A. Krener, "Nonlinear Controllability and Observability," *IEEE Transactions on Automatic Control*, pp. 728–740, 1977.

[160] A. Isidori, *Nonlinear Control Systems.* Springer-Verlag New York, Inc., 3rd ed., 1995.

[161] E. D. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems.* New York, NY, USA: Springer-Verlag New York, Inc., 1998.

[162] Y. Kawano and T. Ohtsuka, "Observability analysis of nonlinear systems using pseudo-linear transformation," in *IFAC Symposium on Nonlinear Control Systems, NOLCOS 2013, Toulouse, France, September 4-6, 2013.*, pp. 606–611, 2013.

[163] M. Travers and H. Choset, "Use of the Nonlinear Observability Rank Condition for Improved Parametric Estimation," in *2015 IEEE International Conference on Robotics and Automation*, pp. 1029–1035, 2015.

[164] R. F. Stengel, "Optimal Control and Estimation (originally published as Stochastic Optimal Control; Theory and Application, Wiley & Sons, New York, 1986," in *Dover Publications*, New York, 1994.

[165] W. Breckenridge, "Quaternions proposed standard conventions," *NASA Jet Propulsion Laboratory, Technical Report*, 1979.

[166] J. Sola, "Quaternion kinematics for the error-state Kalman filter," *CoRR*, vol. abs/1711.02508, 2017.

Chapter 8
Appendix A: Stability Criteria

8.1 Zero Moment Point

In legged robotics, the Zero Moment Point (ZMP) is a characteristic point that is commonly employed not only to infer the gait stability but also for motion generation and feedback control.

The ZMP is derived from the Newton-Euler equations of motion that describe the centroidal dynamics (section 9.1 of appendix B). The latter are formulated as:

$$m\ddot{\boldsymbol{c}} = m\boldsymbol{g} + \boldsymbol{f} \tag{8.1}$$

$$\dot{\boldsymbol{L}}_O = \vec{OC} \times m\mathbf{g} + \boldsymbol{\tau} \tag{8.2}$$

where m is the total mass of the robot, $\boldsymbol{g}$ the gravity vector, C the Center of Mass (CoM) of the system, $\ddot{\boldsymbol{c}}$ the CoM acceleration and $\dot{\boldsymbol{L}}_O$ the rate-of-change of the angular momentum taken at a fixed point O. On the right-hand side, $\boldsymbol{f}$, $\boldsymbol{\tau}$ denotes the contact force/torque respectively, i.e. the sum of all contact forces/torques exerted on the robot, with coordinates taken at the CoM C. An illustration of the latter quantities is given in Figure 8.1.

Subsequently, define the gravito-inertial wrench of the robot, which only depends on its accelerations:

$$\boldsymbol{f}^{gi} = m\left(\boldsymbol{g} - \ddot{\boldsymbol{c}}\right) \tag{8.3}$$

$$\boldsymbol{\tau}_O^{gi} = \vec{OC} \times m\boldsymbol{g} - \dot{\boldsymbol{L}}_O \tag{8.4}$$

ZMPs are the points Z belonging to the non-central axis defined by:

$$\boldsymbol{\tau}_Z^{gi} \times \boldsymbol{n} = \mathbf{0} \tag{8.5}$$

where $\boldsymbol{n}$ is the normal vector of the contact surface, i.e. the ground surface when the robot is walking on flat terrain. To this end, the ZMP can be formally defined as:

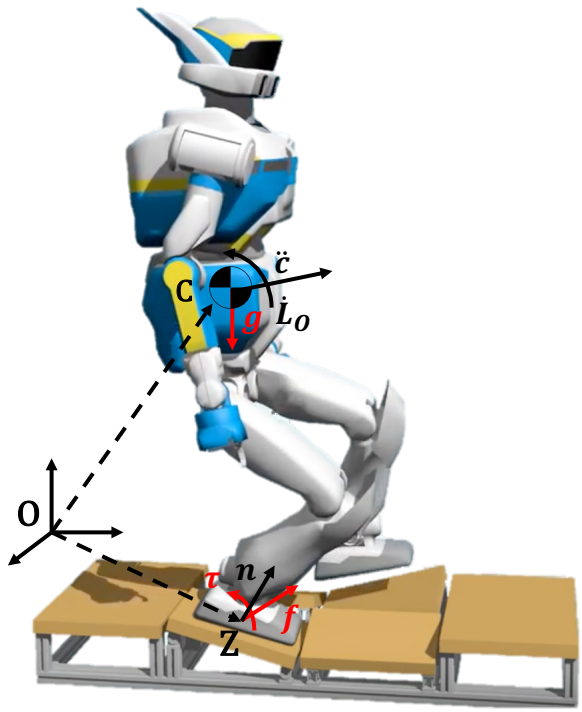

Figure 8.1: HRP-2 walking on uneven terrain [147]. External forces and torques are illustrated in red.

Definition 1 (Zero Moment Point) *The point on the plane around which the horizontal rotation momenta vanish is called the Zero Moment Point (ZMP).*

The left-hand side of (8.5) can be rewritten with respect to the moment taken at another point O:

$$\boldsymbol{\tau}_Z^{gi} \times \boldsymbol{n} = \left(\boldsymbol{\tau}_O^{gi} + \vec{ZO} \times \boldsymbol{f}^{gi}\right) \times \boldsymbol{n} \tag{8.6}$$

$$= \boldsymbol{\tau}_O^{gi} \times \boldsymbol{n} + \left(\vec{ZO} \cdot \boldsymbol{n}\right) \boldsymbol{f}^{gi} - \left(\boldsymbol{f}^{gi} \cdot \boldsymbol{n}\right) \vec{ZO} \tag{8.7}$$

Let us suppose for now that O and Z lie in the same plane normal to $\boldsymbol{n}$, so that $\vec{ZO} \cdot \boldsymbol{n} = 0$. Then, injecting the expression above into the definition of the ZMP yields:

$$\vec{OZ} = \frac{\boldsymbol{n} \times \boldsymbol{\tau}^{gi}}{\boldsymbol{f}^{gi} \cdot \boldsymbol{n}} \tag{8.8}$$

This formula is used in practice to compute the ZMP with Force/Torque (F/T) sensors or with an Inertial Measurement Unit (IMU).

Definition 2 (Quasi-static Stability) *If the ZMP remains within the convex hull created by*

the foot/feet, then the humanoid robot is quasi-statically stable.

The Center of Pressure (COP) is a dynamic point defined in between two bodies in contact. Unlike the ZMP, which we defined from the accelerations of the multi-body system, the COP is a local quantity defined from interaction forces at the contact surface. The COP is equivalent to the ZMP only when the robot is in single contact with the environment and the contact is planar.

8.2 Foot Rotation Indicator

In the previous section, the ZMP was defined as a characteristic point that exists only inside the support polygon during quasi-statically stable walking. Alternatively, one could ask *which is the point on the contact surface where the net ground-reaction force would have to act to prevent foot rotation?* This idea was investigated by A. Goswami [2] and coined as the Foot Rotation Indicator Point (FRI). The latter is formally defined as:

Definition 3 (Foot Rotation Indicator Point) *The point on the foot/ground contact surface, within or outside the convex hull of the foot-support area, at which the resultant moment of the force/torque imposed on the foot is normal to the surface, is the Foot Rotation Indicator (FRI) point.*

Some useful properties of the FRI point which may be exploited in gait planning include the following:

- The FRI point indicates the 'occurrence' of foot rotation, as previously noted.
- The location of the FRI point indicates the 'magnitude' of the unbalanced moment on the foot.
- The FRI point indicates the 'direction' of foot rotation.
- The FRI point indicates the 'stability margin' of the robot, which may be quantified as the minimum distance of the support-polygon boundary from the current location of the FRI point within the footprint. Conversely, when the FRI point is outside the footprint, this minimum distance is a measure of instability of the robot. An imminent foot rotation will be indicated by a motion of the FRI point towards the support-polygon boundary.

8.2.1 Contact Wrench Cone

The contact wrench cone (CWC) is the friction cone of the net contact wrench acting on the robot in multi-contact, i.e. the sum of all contact wrenches as illustrated in Figure 8.2.

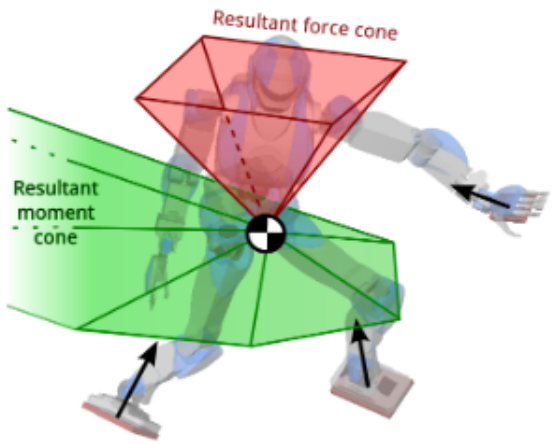

Figure 8.2: CWC is represented by a red force cone and a green moment cone. This is a drawing convenience: in practice, the CWC is a 6D cone where force and moment are not independent. If you choose a resultant force in the red cone, it will affect the shape of the green one [148].

In 2006, H. Hirukawa and coworkers proposed a universal stability criterion of the foot contact of humanoid robots [3]. The proposed method checks whether the sum of the gravity and the inertia wrench applied to the CoM of the robot is inside the CWC. The criterion can be used to determine the strong stability of the foot contact when a robot walks on an arbitrary terrain (even rough terrain) and/or when the hands of the robot are in contact with it under the sufficient friction assumption. This procedure is equivalent to checking whether the ZMP is inside the support polygon of the feet when the robot walks on a horizontal plane with sufficient friction. Finally, when the friction follows a physical law, the criterion can also be used to determine whether the foot contact is weakly stable. Therefore, the proposed criterion can be used to judge the behavior of the ZMP in more general cases.

Definition 4 (Strong Stability Criterion) *If* $(-\boldsymbol{f}, -\boldsymbol{\tau})$ *is an internal element of the polyhedral convex cone of the CWC, then the contact is strongly stable within* $(\boldsymbol{f}, \boldsymbol{\tau})$.

Nevertheless, the strong stability criterion poses some limitations. Especially in the case of motion planning where the sufficient friction assumption is commonly lifted it is not possible to determine the strong stability and thus the generated patterns may not be feasible in the physical world. To this end, the 'weak stability criterion' can be used to check if the planned motions are feasible, but it does not state if the contacts are stable. An alternative idea is to judge if $(-\boldsymbol{f}, -\boldsymbol{\tau})$ should be included in a proper subset of the

polyhedral convex cone of the contact wrench. Then, the contact is likely to be stable within a margin, but there is no guarantee that it is stable within the entire polyhedral convex cone of the contact wrench. The idea is summarized as follows:

Definition 5 (Weak Stability Criterion) *If* $(-\boldsymbol{f}, -\boldsymbol{\tau})$ *is an element of a proper subset of the polyhedral convex cone of the CWC, the contact is called sufficiently weakly stable to* $(\boldsymbol{f}, \boldsymbol{\tau})$.

Subsequently, when additional constraints are imposed on the centroidal motion, the 6D centroidal wrench cone reduces to lower-dimensional areas and volumes that can be used for planning or control:

- When the robot is not moving, contact stability is characterized by the static equilibrium polygon [149] the configuration of the robot is feasible (sustainable) if and only if the CoM lies in a specific polygon, which can be computed efficiently.
- When the robot moves in the Linear Inverted Pendulum Mode (LIPM) [39], i.e. with conserved angular momentum and the CoM constrained on a plane, the CWC reduces to a ZMP support area [149].
- When the robot moves with a conserved angular momentum ($\dot{\boldsymbol{L}} = 0$), the CWC reduces to a 3D cone over CoM accelerations that can be used e.g. for multi-contact locomotion [150].

8.3 Capturability-Based Analysis

Wieber [6] used the concept of the Viability theory [5], to reason about the subset of state space in which the legged system must be maintained to avoid falling. He derived a Lyapunov stability analysis for standing on non-flat terrain given a balance control law. However, the standing assumption precludes the use of this method in walking, and it provides no information on choosing step locations to avoid falling. Moreover, the Viability margin is difficult to compute, limiting its usefulness.

Nevertheless, the Viability theory is closed related to the Capturability theory proposed by Koolen et.al [4] and Pratt et. al [151], which focus mainly on states that are most relevant to normal walking and also provides a method to explicitly compute stepping regions.

For our analysis, let the hybrid system dynamics described by:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad h_i(\mathbf{x}) \neq 0 \tag{8.9}$$

$$\mathbf{x} \leftarrow \mathbf{g}_i(\mathbf{x}), \quad h_i(\mathbf{x}) = 0 \tag{8.10}$$

$$\mathbf{u} \in \mathbf{U}(\mathbf{x}) \tag{8.11}$$

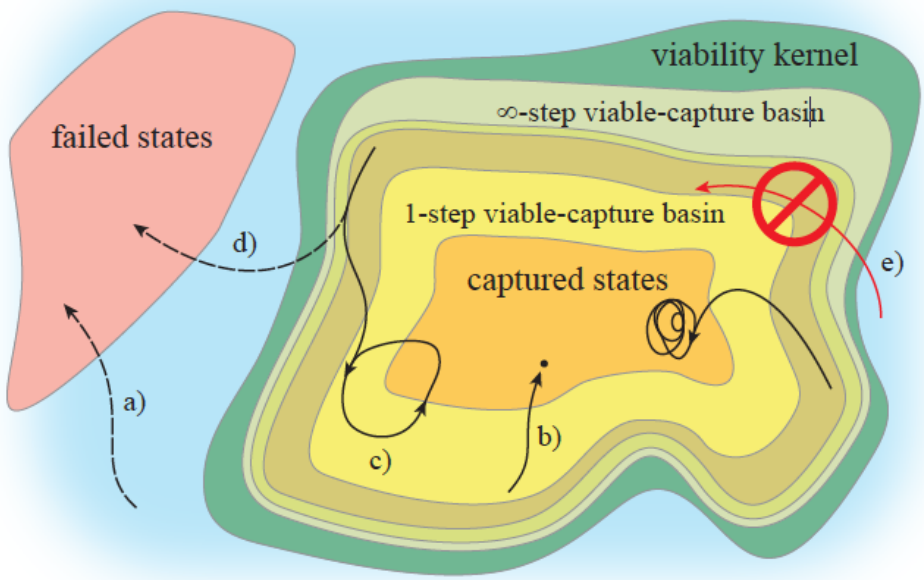

Figure 8.3: State space of a hybrid dynamic system. N-step viable-capture basins are shown. The boundary between two N-step viable-capture basins are shown. The boundary between two N-step viable-capture basins is part of a step surface. The ∞-step viable-capture basin approximates the viability kernel. Five different trajectories are shown: a) a trajectory starting outside the viability kernel inevitably ends up in the set of failed states; b) the system starts in the 1-step viable-capture basin, takes a step, and comes to a rest at a fixed point inside the set of captured states (i.e. the 0-step viable-capture basin); c) a trajectory that eventually converges to a limit cycle; d) a trajectory that has the same initial state as c), but ends up in the set of failed states because the input $\mathbf{u}$ was different; e) impossible trajectory: by definition, it is impossible to enter the viability kernel if the initial state is outside the viability kernel [4].

for $i \in \mathbf{I} \subset \mathbb{N}$, $\mathbf{x}$ is the state of the system and $\mathbf{u}$ is the control input and is confined to the state-depended set of allowable control input $\mathbf{U}(\mathbf{x})$. When the system state lies on the switching surface, $h_i(\mathbf{x}) = 0$, the discrete jump dynamics reset the state to $\mathbf{g}_i(\mathbf{x})$ instantaneously. Koolen et.al [4] assumed that some part of the state space must be avoided at all cost, namely the set of failed states, for humanoid walking this is the set with all states that lead to a fall. The viability kernel proposed by Aubin et al. [5] and introduced in the field of legged locomotion [6], is the set of all states from which these failed states can be

avoided. To this end, for every initial state inside the viability kernel, there exists at least one trajectory that can never converge to a failed state. As long as the system state remains within its viability kernel, the system characterized as viable.

The viability theory is a very natural concept and can be seen as a very generic definition of stability for a dynamic system. However, determining the viability kernel is generally analytically intractable, and approximation is computationally expensive as seen in Wieber's work [6]. Furthermore, it is far from trivial to design a controller based solely on the viability kernel. Towards that direction, Koolen et.al [4], proposed the N-step capturability, which is a more restrictive definition of walking stability. N-step capturability dictates that the legged system must come to a stop by taking exactly N or fewer steps.

Definition 6 (N-step Capturable) *Let $\mathbf{X}_{\text{failed}}$ be the set of failed states associated with the hybrid dynamic system given in Eq.(8.9). A state $\mathbf{x}_0$ of this system is N-step capturable with respect to $\mathbf{X}_{\text{failed}}$, for $N \in \mathbb{N}$, if and only if there exists at least one trajectory starting at $\mathbf{x}_0$ that contains N or fewer crossings of switching surfaces (steps), and never reaches $\mathbf{X}_{\text{failed}}$.*

Inspired by the viable-capture basin [152] defined with the viability kernel, the authors defined the N-step viable-capture basin as the set of all N-step capturable states. The 0-step viable-capture basin was referred to as the set of captured states, and if a system's state is within the 0-step viable-capture basin, the system was referred to as captured. The N-step viable-capture basins, illustrated in Figure 8.3, describe the subsets of state space which can be achieved by a control input $\mathbf{u}$ and the system can come to a stop (a captured state) by taking N or fewer steps.

For $N > 0$ the N-step viable-capture basin is equivalent to the set containing every initial state $\mathbf{x}_0$ for which at least one trajectory containing a single step and starting from $\mathbf{x}_0$ reaches the $(N-1)$-step viable-capture basin in finite time, while never reaching a failed state. This property allow us to recursively compute the N-step viable-capture basin.

As Koolen et.al [4] pointed out, the N-step viable-capture basins as well as the viability kernels do not provide a direct means of controller design. This motivated them to introduce the notion of N-step capture points and N-step capture regions. Capture points and capture regions are defined in the Euclidean space, contrasted to the viability kernel and the viable-capture basin that are defined in the state-space. Subsequently, they describe places where the legged robot can step to to come to a stop (reach a captured state). Figure 8.4 shows the N-capture regions for illustration purposes.

Definition 7 (N-step capture point, region) *Let $\mathbf{x}_0$ be the state of a hybrid dynamic system defined by Eq.(8.9), with an associated set of failed states $\mathbf{X}_{\text{failed}}$. A point $\mathbf{r}$, is an N-step capture point for this system, for $N > 0$, if and only if there exists at least one trajectory starting at $\mathbf{x}_0$ that contains one step, never reaches $\mathbf{X}_{\text{failed}}$, reaches an $(N-1)$-step capturable*

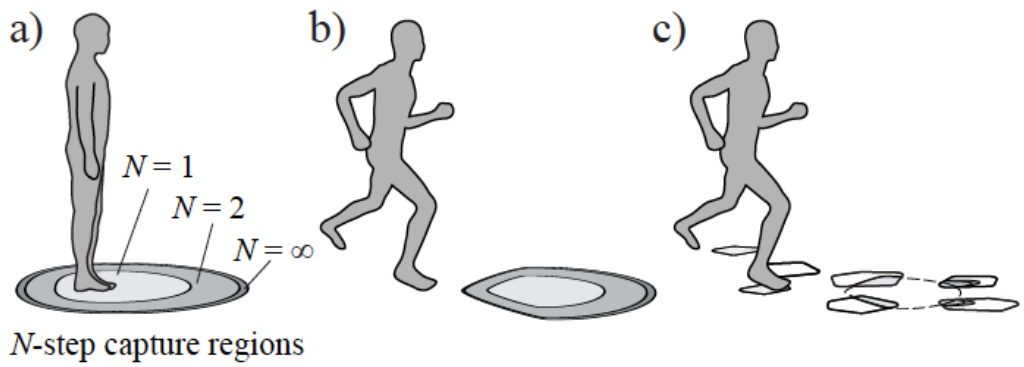

Figure 8.4: a) A conceptual representation of the N-step capture regions for a human in a captured state (standing at rest). b) N-step capture regions for a running man. The capture regions have decreased in size and have shifted, as compared to a). c) N-step capture regions for the same state as b), but with sparse footholds (e.g. stepping stones in a pond). The set of failed states has changed, which is reflected in the capture regions [4].

state, and places a contact reference point at $\mathbf{r}$ *at the time of the step. The* $N - step$ *capture region is the set of all* N*-step capture points.*

8.4 The Gait Sensitivity Norm

So far, we presented various stability measures, that were based on the assumption that a biped can prevent a fall by placing its foot within an appropriate way to the ground. These measures, indicate how close a humanoid is to tipping by measuring the distance from the edge of the support foot to the projection of the center of mass (the static stability margin) or the COP (the ZMP stability margin [153] or the FRI [2]).

Nevertheless, continuous flat foot contact is not necessary to prevent falling, this is proven with the *limit cycle walkers*, e.g bipeds that show stable limit cycle motion without having local controllability at all times during the gait. Such an example is the passive walker by McGeer [154]. His biped was equipped with arc-shaped feet, which make it impossible to achieve local controllability at any point in time, and nonetheless, it showed perfectly stable gait.

For limit cycle walkers, worldwide there exist three additional measures to quantify disturbance rejection. First, the Basin of Attraction (BoA), is the total set of system states of a limit cycle walker for which its gait converges to its nominal limit cycle [155]. To compute

the BoA can either be computationally expensive if a full evaluation of the nonlinear system behavior is done or hard to find if a Lyapunov stability analysis is followed, since the 'correct' Lyapunov function need to be found in advance. However, the benefit of the BoA is the good correlation between the actual disturbance rejection and the distance from the limit cycle to the borders of the BoA. Another measure is the largest Floquet multiplier. Floquet multipliers indicate how fast small deviations from the limit cycle converge on a step-to-step basis [156]. For a stable limit cycle, the Floquet multipliers have to be within the unit circle, the closer to zero, the faster the convergence rate. The benefit of the Floquet multipliers is that they require a short computation time as they involve only small deviations from the limit cycle. The drawback is the limited correlation between the actual disturbance rejection and the distance from the largest Floquet multiplier to the unit circle [155]. Finally, some researcher have measured the disturbance rejection of the limit cycle walkers by measuring the largest deterministic disturbance a biped can handle without tipping over. This measure obviously shows good correlation with the disturbance rejection properties, however, it requires long experimentation.

Hobbelen and Wisse [7] introduced the Gait Sensitivity Norm (GSN) $\left\|\frac{\partial \mathbf{g}}{\partial \mathbf{e}}\right\|_2$. This measure is used to quantify the effect of a set of disturbances on a walking gait. Let $\mathbf{e}$ be a set of disturbances (system inputs) and $\mathbf{g}$ be a set of gait indicators (system outputs). The GSN measures the size of the dynamic response of this system. The selection of the disturbance set $\mathbf{e}$ and the gait indicators $\mathbf{g}$ are open parameters to the designer, however they need to have expert knowledge to this measure, i.e possible disturbances $\mathbf{e}$ can be the motion once per step, such as floor irregularities, and continuous disturbances such as sensor noise or torque ripple, while possible gait indicators $\mathbf{g}$ can be step width, step time or ground clearance at midswing.

The authors compared the GSN with the Floquet multipliers and the largest allowable single disturbance measure using a simple $2D$ walking model. They used floor irregularities as disturbance $\mathbf{e}$ and step time as indicator for the chance of falling $\mathbf{g}$. The reciprocal of the GSN $(1/\left\|\frac{\partial \mathbf{g}}{\partial \mathbf{e}}\right\|_2)$, was faster in computation time and better in correlation with the actual disturbance rejection, contrasted to the maximum Floquet multiplier distance from the unit cycle $(1-\max(|\lambda|))$ and the largest allowable deterministic disturbance $(\max(|e|))$. Moreover, they obtained similar results by the same comparison on the 'Meta' physical prototype. Nonetheless, the experiment indicates that the measured Gait Sensitivity Norm on the real prototype gives a good prediction of the actual disturbance rejection.

Although this is a systematic attempt to quantify the stability of the limit cycle walkers and provide a better insight on the stability of the walking limit cycle, the analysis is strictly based on the experience and the knowledge of the designer in the walking system, since the set of $\mathbf{e}$ of disturbances and $\mathbf{g}$ of the gait indicators must be defined.

8.5 Summary

In the previous sections, five popular stability analysis approaches, which received attention by researchers worldwide, were introduced.

Initially, the well-known Zero Moment Point (ZMP) criterion, which is most often applied in humanoid robots, was presented.

- Quasi-static stability of a humanoid robot is maintained, while the ZMP remains inside the convex hull created by the support foot/feet.
- The ZMP realized and continues to realize an essential role in both theoretical considerations and practical development of humanoid and biped locomotion.

Next, the Foot Rotation Indicator (FRI) point was briefly presented, providing a deeper insight into the humanoid robot stability.

- The farther away the FRI point is from the support boundary, the larger the unbalanced moment on the foot and the greater the instability.
- The distance between the FRI point and the nearest point on the polygon boundary is a useful indicator of the static stability margin of the foot.
- The ZMP/FRI point is extensively used for gait planning and control.

Subsequently, the Contact Wrench Cone (CWC) universal stability criterion was briefly described. This criterion checks if the sum of the gravity and the inertia wrench applied to the CoM of a humanoid robot is inside the polyhedral convex cone of the contact wrench between the feet of the robot and its environment.

- Possible to determine strong stability of the foot contact even when a robot walks on an arbitrary terrain and/or when the hands of the robot are in contact.
- While walking on horizontal flat terrain the criterion reduces to the ZMP criterion.
- CWC constraints can be enforced in gait planning and control.

Following, the Capturability theory was presented, based on the general idea of the Viability theory. The Captured States were defined as the states were the robot can come to a complete stop by stepping on the capture point, maintaining its balance.

- More efficient to compute than the Viability kernel.
- Capture regions and Capture Points can be used for gait planning and control.

Finally, the Gait Sensitivity Norm (GSN) was presented as a metric to quantify the stability of the limit cycle walkers under a particular control law.

- More efficient in terms of computational time and good disturbance-rejection correlation contrasted to the BoA, the Floquet multipliers, and the largest allowable deterministic disturbance.
- Strong dependence to the input disturbance set and the output gait indicators that must be pre-determined by the designer experience.
- No direct implications to gait planning and control.

Chapter 9
Appendix B: Mathematical Formulations and Derivations

9.1 Newton-Euler Equations of Motion

In this section, the Newton-Euler equations of motion for the six unactuated DoFs of walking robot are derived. The final result is:

$$\dot{\boldsymbol{P}} = m\boldsymbol{g} + \sum_{\text{contact i}} \boldsymbol{f}_i \tag{9.1}$$

$$\dot{\boldsymbol{L}} = \sum_{\text{contact i}} (\boldsymbol{p}_i - \boldsymbol{c}) \times \boldsymbol{f}_i + \boldsymbol{\tau}_i \tag{9.2}$$

9.1.1 Newton's Equation

Let $\boldsymbol{p}_i$ denote the position with respect to the world frame of a reference point on the robot's i^{th} link. Furthermore, let $\boldsymbol{c}$ denote the Center of Mass (CoM) and m_i the mass of link i with $m = \sum_{\text{link i}} m_i$ the total mass.

While walking, the robot is subject to gravity and contact forces. Let $\boldsymbol{g}$ denote the gravity vector and for each link i in contact with the environment $\boldsymbol{f}_i$ represent the resultant force exerted on the link. Subsequently, let $\boldsymbol{h}_{ij}$ denote the internal force exerted by link i on link j. By convention $\boldsymbol{h}_{ij} = 0$ if link i and j are not connected and $\boldsymbol{f}_i = 0$ if link i is not in contact with the environment.

Newton's equation of motion relates the resultant accelerations and forces:

$$\sum_{\text{link i}} m_i \ddot{\boldsymbol{p}}_i = \sum_{\text{link i}} m_i \boldsymbol{g} + \boldsymbol{f}_i + \sum_{\text{link j}\neq\text{i}} \boldsymbol{h}_{ij} \tag{9.3}$$

Newton's third law of motion states that $\boldsymbol{h}_{ij} = -\boldsymbol{h}_{ji}$. To this end all internal forces vanish:

$$\sum_{\text{link i}} \sum_{\text{link j}\neq\text{i}} \boldsymbol{h}_{ij} = 0 \tag{9.4}$$

In concise form, Newton's equation binds the acceleration of the CoM with the whole body resultant force. Thus, let $\boldsymbol{P}$ denote the linear momentum of the robot defined by:

$$\boldsymbol{P} = m\dot{\boldsymbol{c}} \tag{9.5}$$

Consequently, Newton's equation can be re-written as:

$$\dot{\boldsymbol{P}} = m\boldsymbol{g} + \sum_{\text{contact i}} \boldsymbol{f}_i \tag{9.6}$$

stating that the rate of change of the linear momentum is equal to the resultant of external forces exerted on the robot.

9.1.2 Euler's Equation

As described above, the Newton's equation binds the translations of the robot's links. Euler's equation provides a similar binding for the links orientation.

Let $\boldsymbol{R}_i$ denote the rotation matrix from the link frame i to the world frame of reference. In addition, let $\boldsymbol{\omega}_i$ denote is angular velocity and $\boldsymbol{I}_i$ be the inertia matrix taken at the center of mass of the link, both expressed in the world frame.

For a link i in contact with the environment, we define $\boldsymbol{\tau}_i$ to be the resultant moment of contact forces exerted on the link at the reference point $\boldsymbol{p}_i$. If the link is in point contact with the environment at $\boldsymbol{p}_i$, the moment will be zero. On the contrary, if the link is in planar contact, both $\boldsymbol{f}_i$ and $\boldsymbol{\tau}_i$ are non-zero.

Euler's equation of motion links the angular momenta and resultant moments of external forces:

$$\sum_{\text{link i}} (\boldsymbol{p}_i - \boldsymbol{c}) \times m_i\ddot{\boldsymbol{p}}_i + \boldsymbol{I}_i\dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times (\boldsymbol{I}_i\boldsymbol{\omega}_i) = \sum_{\text{link i}} (\boldsymbol{p}_i - \boldsymbol{c}) \times (\boldsymbol{f}_i + m_i\boldsymbol{g}) + \boldsymbol{\tau}_i \tag{9.7}$$

Nevertheless, by a similar argument to the vanishing of internal forces in the translational case, moments of internal forces do not appear in (9.7). Furthermore, by the definition of the CoM, $\sum_{\text{link i}} (\boldsymbol{p}_i - \boldsymbol{c}) \times m_i\boldsymbol{g} = 0$.
The angular momentum of the robot is defined as:

$$\boldsymbol{L} = \sum_{\text{link i}} (\boldsymbol{p}_i - \boldsymbol{c}) \times m_i\dot{\boldsymbol{p}}_i + \boldsymbol{I}_i\boldsymbol{\omega}_i \tag{9.8}$$

Then, Euler's equation can be formulated in concise form as:

$$\dot{\boldsymbol{L}} = \sum_{\text{contact i}} (\boldsymbol{p}_i - \boldsymbol{c}) \times \boldsymbol{f}_i + \boldsymbol{\tau}_i \tag{9.9}$$

stating that the rate of change of the angular momentum is equal to the resultant moment of the external forces exerted on the robot.

9.2 Leg Odometry

Humanoids locomote by alternating their supporting feet on the ground. The forward kinematics along with the time derivatives of the base and the supporting foot are given by:

$$
\begin{aligned}
{}^{w}\boldsymbol{p}_s &= {}^{w}\boldsymbol{p}_b + {}^{w}\boldsymbol{R}_b{}^{b}\boldsymbol{p}_s && (9.10)\\
{}^{w}\boldsymbol{v}_s &= {}^{w}\boldsymbol{v}_b + {}^{w}\boldsymbol{\omega}_b \times {}^{w}\boldsymbol{R}_b{}^{b}\boldsymbol{p}_s + {}^{w}\boldsymbol{R}_b{}^{b}\boldsymbol{v}_s && (9.11)\\
{}^{w}\boldsymbol{R}_s &= {}^{w}\boldsymbol{R}_b{}^{b}\boldsymbol{R}_s && (9.12)\\
{}^{w}\boldsymbol{\omega}_s &= {}^{w}\boldsymbol{\omega}_b + {}^{w}\boldsymbol{R}_b{}^{b}\boldsymbol{\omega}_s && (9.13)
\end{aligned}
$$

where the notation is according to chapter 3.

Commonly, ${}^{b}\boldsymbol{p}_s$, ${}^{b}\boldsymbol{v}_s$, ${}^{b}\boldsymbol{R}_s$, and ${}^{b}\boldsymbol{\omega}_s$, are computed with forward kinematics, while ${}^{w}\boldsymbol{R}_b$ and ${}^{w}\boldsymbol{\omega}_b$ can be accurately estimated with a complementary filter and IMU measurements [51, 52].

Under the assumption that the supporting foot is stationary and in contact with the environment during a step, ${}^{w}\boldsymbol{p}_s$ can be initially defined and ${}^{w}\boldsymbol{v}_s$ is zero. Subsequently, ${}^{w}\boldsymbol{p}_b$ and ${}^{w}\boldsymbol{v}_b$ can be derived from (9.10) and (9.11) as:

$$
\begin{aligned}
{}^{w}\boldsymbol{p}_b &= {}^{w}\boldsymbol{p}_s - {}^{w}\boldsymbol{R}_b{}^{b}\boldsymbol{p}_s && (9.14)\\
{}^{w}\boldsymbol{v}_b &= -{}^{w}\boldsymbol{\omega}_b \times {}^{w}\boldsymbol{R}_b{}^{b}\boldsymbol{p}_s - {}^{w}\boldsymbol{R}_b{}^{b}\boldsymbol{v}_s && (9.15)
\end{aligned}
$$

Nevertheless, in dynamic locomotion the supporting foot commonly rotates about the contact points, rolls over the ground, or even loses contact with it. To this end, the previous assumption does not hold true under all circumstances. In order to improve the accuracy of leg odometry, Masuya et al. [157, 158] proposed a dead reckoning method based on the Anchoring Pivot (AP), defined inside the supporting foot. The forward kinematics between the AP and the supporting foot are:

$$
\begin{aligned}
{}^{w}\boldsymbol{p}_{sa} &= {}^{w}\boldsymbol{p}_s + {}^{w}\boldsymbol{R}_s{}^{s}\boldsymbol{p}_{sa} && (9.16)\\
{}^{w}\boldsymbol{v}_{sa} &= {}^{w}\boldsymbol{v}_s + {}^{w}\boldsymbol{\omega}_s \times {}^{w}\boldsymbol{R}_s{}^{s}\boldsymbol{p}_{sa} && (9.17)
\end{aligned}
$$

since ${}^{s}\boldsymbol{v}_{sa} = 0$.

When ${}^{s}\boldsymbol{p}_{sa}$ is computed it provides the current ${}^{w}\boldsymbol{v}_{sa}$, which is also the pivot at the next time instance, as illustrated in Figure 9.1. This is true under the assumption that ${}^{w}\boldsymbol{v}_{sa} \approx 0$

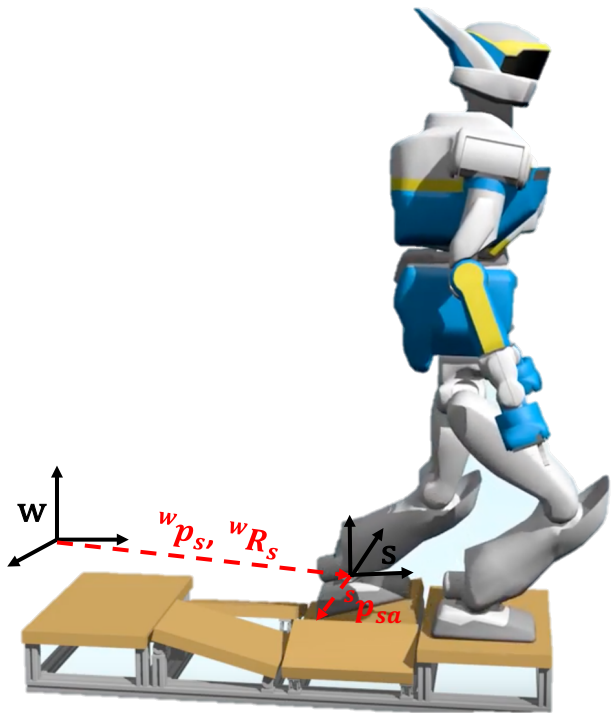

Figure 9.1: HRP-2 walking on uneven terrain [147]. Updating the support foot pose based on the AP.

and ${}^{w}\boldsymbol{p}_{sa}$ is invariant momentarily. Consequently, ${}^{w}\boldsymbol{p}_s$ and ${}^{w}\boldsymbol{v}_s$ can be obtained as:

$$ {}^{w}\boldsymbol{p}_s = {}^{w}\boldsymbol{v}_{sa} - {}^{w}\boldsymbol{R}_s{}^{s}\boldsymbol{v}_{sa} \tag{9.18} $$

$$ {}^{w}\boldsymbol{v}_s = -{}^{w}\boldsymbol{\omega}_s \times {}^{w}\boldsymbol{R}_s{}^{s}\boldsymbol{v}_{sa} \tag{9.19} $$

To this end, ${}^{w}\boldsymbol{p}_b$ and ${}^{w}\boldsymbol{v}_b$ can be derived with (9.14) and (9.15) respectively.

Let us consider the case where the left foot is the support foot. In order to compute the ${}^{l}\boldsymbol{p}_{la}$, the following hypotheses are employed:

- AP is the point with the instantaneous minimum velocity with respect to the support frame, e.g. ${}^{l}\boldsymbol{v}_{la} \approx 0$.
- AP is invariant for a short amount of time namely, ${}^{l}\delta\boldsymbol{p}_{sa} \approx 0$
- AP is located near the point of action of the total GRF.

The above hypotheses can be mathematically formulated as:

$$ \min_{{}^{l}\boldsymbol{p}_{la}} J[k] = J[k]_0 + a_1 J[k]_1 + a_2 J[k]_2 \tag{9.20} $$

where k is the k-th discrete time instant, a_1, a_2 are optimization weights, and

$$J[k]_0 = \frac{1}{2\zeta_0}\|\delta^l \boldsymbol{p}_{la}\|^2 \tag{9.21}$$

$$J[k]_1 = \frac{1}{2}\|{}^w\boldsymbol{v}_l + {}^w\boldsymbol{\omega}_l \times {}^w\boldsymbol{R}_l{}^l\boldsymbol{p}_{la}\|^2 \tag{9.22}$$

$$J[k]_2 = \frac{1}{2\zeta_1}\|{}^l\boldsymbol{\tau}_l + \left({}^l\boldsymbol{p}_{lf} - {}^l\boldsymbol{p}_{la}\right) \times {}^l\boldsymbol{f}_l\|^2 \tag{9.23}$$

with ${}^l\boldsymbol{p}_{lf}$ be the point of action of the total GRF in the left foot and ζ_0, ζ_1 be dimension transformation constants e.g. $\zeta_0 = dt^2$ and $\zeta_1 = (mgdt)^2$.

Subsequently, the left foot odometry can be updated as:

$${}^w\boldsymbol{p}_l[k] = {}^w\boldsymbol{p}_l[k-1] + {}^w\boldsymbol{R}_l[k-1]{}^l\boldsymbol{p}_{la}[k] - {}^w\boldsymbol{R}_l[k]{}^l\boldsymbol{p}_{la}[k] \tag{9.24}$$

and the base with respect to the left foot is:

$${}^w\boldsymbol{p}_{bl}[k] = {}^w\boldsymbol{p}_l[k] + +{}^w\boldsymbol{R}_b[k]{}^b\boldsymbol{p}_l[k] \tag{9.25}$$

Likewise, the same quantities can be computed for the right foot.

Finally, the leg odometry ${}^w\boldsymbol{p}_b$ is derived as a weighted sum for both feet namely:

$${}^w\boldsymbol{p}_b = w_l{}^w\boldsymbol{p}_{bl} + w_r{}^w\boldsymbol{p}_{br} \tag{9.26}$$

with

$$w_l = \frac{{}^w\boldsymbol{f}_l^z + \epsilon}{{}^w\boldsymbol{f}_l^z + {}^w\boldsymbol{f}_r^z + 2\epsilon} \tag{9.27}$$

$$w_r = \frac{{}^w\boldsymbol{f}_r^z + \epsilon}{{}^w\boldsymbol{f}_l^z + {}^w\boldsymbol{f}_r^z + 2\epsilon} \tag{9.28}$$

where ϵ is a small positive constant to avoid division by zero.

9.3 Nonlinear Observability

Consider the following nonlinear dynamical system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{9.29}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) \tag{9.30}$$

with $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathbb{R}^l$ is the input vector and $\boldsymbol{y} \in \mathbb{R}^m$ is the measured output. In addition, without the loss of generality we assume that $\boldsymbol{f}$ and $\boldsymbol{h}$ are smooth functions. The general question is under which conditions we are able to reconstruct the state vector $\boldsymbol{x}$ by observ-

ing the system's output $\boldsymbol{y}$. There are many results on nonlinear observability i.e [159–162], nevertheless, we will follow the results presented by Del Vecchio and Murray [88], and effectively used in [163], since our output dynamics are depended on the input $\boldsymbol{u}$.

Let $\boldsymbol{h}(\boldsymbol{x},\boldsymbol{u}) = \Big[h_1(\boldsymbol{x},\boldsymbol{u}),\ldots,h_m(\boldsymbol{x},\boldsymbol{u})\Big]^\top$ and $\bar{\boldsymbol{u}} = \Big[u_1,\ldots,u_1^{(n_1-1)},\ldots,u_l,\ldots,u_l^{(n_l-1)}\Big]^\top$ with $\sum_{i=1}^{l} n_i = n_u$.

Subsequently, define the following functions:

$$\varphi_i^0 = h_i \tag{9.31}$$

$$\varphi_i^j = L_f\varphi_i^{j-1} = \frac{\partial\varphi_i^{j-1}}{\partial\boldsymbol{x}}\boldsymbol{f} + \sum_{k=0}^{j-1}\frac{\partial\varphi^{j-1}}{\partial\boldsymbol{u}^{(k)}}\boldsymbol{u}^{(k+1)} = y_i^{(j)} \tag{9.32}$$

where $L_f\varphi_i^{j-1}$ is the Lie derivative of φ_i^{j-1} in the direction of the vector field $\boldsymbol{f}$ and coincides with the j-th derivative of the i-th output, $y_i^{(j)}$.

Next, define the map $\boldsymbol{\Phi}(\boldsymbol{x},\bar{\boldsymbol{u}}) : \mathbb{R}^n \times \mathbb{R}^{n_u} \to \mathbb{R}^n$ to be:

$$\boldsymbol{\Phi}(\boldsymbol{x},\bar{\boldsymbol{u}}) = \Big[h_1,\varphi_1^1,\ldots,\varphi_1^{k_1-1},\ldots,h_m,\varphi_m^1,\ldots,\varphi_m^{k_m-1}\Big]^\top$$

$\forall\, k_i \mid \sum_{i=1}^{m} k_i = n$.

Then, the system in (9.29), (9.30) is *locally observable* if there exits a non-empty set $\mathcal{X}\times\mathcal{U} \subset \mathbb{R}^n \times \mathbb{R}^{n_u}$, such that the map $\boldsymbol{\Phi}(\boldsymbol{x},\bar{\boldsymbol{u}})$, for some k_i, is invertible with respect to $\boldsymbol{x}$ and its inverse is smooth $\forall\ (\boldsymbol{x},\bar{\boldsymbol{u}}) \in \mathcal{X}\times\mathcal{U}$, in other words:

$$\mathrm{rank}\boldsymbol{\mathcal{O}} = \mathrm{rank}\left(\frac{\partial\Phi(\mathrm{x},\bar{\mathrm{u}})}{\partial\mathrm{x}}\right) = \mathrm{n} \tag{9.33}$$

where $\boldsymbol{\mathcal{O}}$ is the local nonlinear observability matrix.

Notice, the choice of coordinates needed to define the map $\boldsymbol{\Phi}$ depend on the dynamics and are not unique, since there are many combination of k_i's that sum up to n. To this end, it suffices to find a map that satisfies the condition in (9.33).

9.4 Cross-Correlation of Disturbance Input and Measurement Noise

Contents of this proof are adopted from section 4.4 Correlated Disturbance Inputs and Measurement Noise, pp. 361-364 of [164] and reproduced following the notation introduced in Chapter 3

Assume the linearized discrete-time system:

$$\boldsymbol{x}_k = \boldsymbol{F}_{\boldsymbol{x}_{k-1}}\boldsymbol{x}_{k-1} + \boldsymbol{F}_{\boldsymbol{u}_{k-1}}\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1}$$
$$\boldsymbol{y}_k = \boldsymbol{H}_{\boldsymbol{x}_k}\boldsymbol{x}_k + \boldsymbol{H}_{\boldsymbol{u}_k}\boldsymbol{u}_k + \boldsymbol{n}_k$$

where $\boldsymbol{x}_k$ is the state, $\boldsymbol{u}_k$ is the input, $\boldsymbol{w}_k$ is zero mean Gaussian noise with covariance $\boldsymbol{Q}_k$ (process noise), $\boldsymbol{y}_k$ is the output, $\boldsymbol{n}_k$ is zero mean Gaussian noise with covariance $\boldsymbol{R}_k$ (measurement noise), and $\boldsymbol{F}_{\boldsymbol{x}_k}$, $\boldsymbol{F}_{\boldsymbol{u}_k}$, $\boldsymbol{H}_{\boldsymbol{x}_k}$, $\boldsymbol{H}_{\boldsymbol{u}_k}$ are the linearizations of $\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ and $\boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ with respect to $\boldsymbol{x}_k$ and $\boldsymbol{u}_k$ respectively.

A linear minimum-variance filter for this system takes the form:

$$\begin{aligned}\hat{\boldsymbol{x}}_k(-) &= \boldsymbol{F}_{\boldsymbol{x}_{k-1}}\hat{\boldsymbol{x}}_{k-1}(+) + \boldsymbol{F}_{\boldsymbol{u}_{k-1}}\boldsymbol{u}_{k-1}\\ \hat{\boldsymbol{x}}_k(+) &= \hat{\boldsymbol{x}}_k(-) + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{H}_{\boldsymbol{x}_k}\hat{\boldsymbol{x}}_k - \boldsymbol{H}_{\boldsymbol{u}_k}\boldsymbol{u}_k)\end{aligned}$$

where $\boldsymbol{K}_k$ is the optimal filter gain to be determined, $\hat{\boldsymbol{x}}_k(-)$ is the state estimate before the measurement update and $\hat{\boldsymbol{x}}_k(+)$ is the estimate after the update. Defining the equivalent state residuals as:

$$\begin{aligned}\boldsymbol{e}_k(-) &= \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k(-)\\ \boldsymbol{e}_k(+) &= \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k(+)\end{aligned}$$

the dynamics of the estimation error can be described by:

$$\boldsymbol{e}_k(-) = \boldsymbol{F}_{\boldsymbol{x}_{k-1}}\boldsymbol{e}_{k-1}(+) + \boldsymbol{w}_{k-1} \tag{9.34}$$

$$\boldsymbol{e}_k(+) = \boldsymbol{e}_k(-) - \boldsymbol{K}_k(\boldsymbol{H}_{\boldsymbol{x}_k}\boldsymbol{e}_k(-) + \boldsymbol{n}_k) \tag{9.35}$$

Thus, the known control input has no effect on the estimation error, therefore $\boldsymbol{H}_{\boldsymbol{x}_k}$ is rewritten as $\boldsymbol{H}_k$ for simplicity in the following.

The disturbance input and measurement noise are modeled as a time-skewed, white joint stochastic process such that:

$$E\left\{\begin{bmatrix}\boldsymbol{w}_{k-1}\\ \boldsymbol{n}_k\end{bmatrix}\begin{bmatrix}\boldsymbol{w}_{k-1}^\top & \boldsymbol{n}_k^\top\end{bmatrix}\right\} = \begin{bmatrix}\boldsymbol{Q}_{k-1} & \boldsymbol{M}_k\\ \boldsymbol{M}_k^\top & \boldsymbol{R}_k\end{bmatrix}$$

and

$$E\begin{bmatrix}\boldsymbol{w}_{k-1}\\ \boldsymbol{n}_k\end{bmatrix} = \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{0}\end{bmatrix}$$

$\boldsymbol{M}_k$ expresses the cross-correlation between the disturbances and the measurement at discrete time k.

The covariances of the state estimate error before and after the measurement update are expressed by forming the outer products of the corresponding state residuals (Eqs.9.34,

9.35) and taking the expected values:

$$
\begin{aligned}
E[\boldsymbol{e}_k(-)\boldsymbol{e}_k(-)^\top] &= \boldsymbol{P}_k(-) \\
&= E[\boldsymbol{F}_{\boldsymbol{x}_{k-1}}\boldsymbol{e}_{k-1}(+)\boldsymbol{e}_{k-1}(+)^\top\boldsymbol{F}_{\boldsymbol{x}_{k-1}}^\top + \boldsymbol{w}_{k-1}\boldsymbol{w}_{k-1}^\top] \\
&= \boldsymbol{F}_{\boldsymbol{x}_{k-1}}\boldsymbol{P}_{k-1}(+)\boldsymbol{F}_{\boldsymbol{x}_{k-1}}^\top + \boldsymbol{Q}_{k-1} \qquad (9.36) \\
E[\boldsymbol{e}_k(+)\boldsymbol{e}_k(+)^\top] &= \boldsymbol{P}_k(+) \\
&= E[(\boldsymbol{e}_k(-) - \boldsymbol{K}_k(\boldsymbol{H}_k\boldsymbol{e}_k(-) + \boldsymbol{n}_k)(\boldsymbol{e}_k(-) - \boldsymbol{K}_k(\boldsymbol{H}_k\boldsymbol{e}_k(-) + \boldsymbol{n}_k)^\top] \\
&= \boldsymbol{P}_k - E[\boldsymbol{e}_k(-)\boldsymbol{e}_k(-)^\top\boldsymbol{H}_k^\top\boldsymbol{K}_k^\top + \boldsymbol{e}_k(-)\boldsymbol{n}_k^\top\boldsymbol{K}_k^\top] \\
&- E[\boldsymbol{K}_k\boldsymbol{H}_k\boldsymbol{e}_k(-)\boldsymbol{e}_k(-)^\top + \boldsymbol{K}_k\boldsymbol{n}_k\boldsymbol{e}_k(-)^\top] \\
&+ E[\boldsymbol{K}_k\boldsymbol{H}_k\boldsymbol{e}_k(-)\boldsymbol{e}_k(-)^\top\boldsymbol{H}_k^\top\boldsymbol{K}_k^\top \\
&+ \boldsymbol{K}_k\boldsymbol{H}_k\boldsymbol{e}_k(-)\boldsymbol{n}_k^\top\boldsymbol{K}_k^\top + \boldsymbol{K}_k\boldsymbol{n}_k\boldsymbol{e}_k^\top(-)\boldsymbol{H}_k^\top\boldsymbol{K}_k^\top \\
&+ \boldsymbol{K}_k\boldsymbol{n}_k\boldsymbol{n}_k^\top\boldsymbol{K}_k^\top] \qquad (9.37)
\end{aligned}
$$

Substituting (9.34) in (9.37) and moving the expectation operation inside the deterministic system matrices, the postupdate covariance can be written in the Joseph form as:

$$
\begin{aligned}
\boldsymbol{P}_k(+) =& (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_k(-)(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)^\top + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^\top + \boldsymbol{K}_k(\boldsymbol{H}_k\boldsymbol{M}_k + \boldsymbol{M}_k^\top\boldsymbol{H}_k^\top)\boldsymbol{K}_k^\top \\
&- \boldsymbol{M}_k\boldsymbol{K}_k^\top - \boldsymbol{K}_k\boldsymbol{M}_k^\top \qquad (9.38)
\end{aligned}
$$

The optimal gain matrix that minimizes the expected value of the state-residual squared at each step:

$$
J_k = E[\boldsymbol{e}_k(+)^\top\boldsymbol{e}_k(+)] = Tr[\boldsymbol{P}_k(+)] \qquad (9.39)
$$

Consequently,

$$
\frac{\partial J_k}{\partial \boldsymbol{K}_k} = 2\left(\boldsymbol{K}_k\left(\boldsymbol{H}_k\boldsymbol{P}_k(-)\boldsymbol{H}_k^\top + \boldsymbol{H}_k\boldsymbol{M}_k + \boldsymbol{M}_k^\top\boldsymbol{H}_k^\top + \boldsymbol{R}_k\right) - \boldsymbol{P}_k(-)\boldsymbol{H}_k^\top - \boldsymbol{M}_k\right) = \boldsymbol{0} \qquad (9.40)
$$

and the optimal gain is:

$$
\boldsymbol{K}_k = \left(\boldsymbol{P}_k(-)\boldsymbol{H}_k^\top + \boldsymbol{M}_k\right)\left(\boldsymbol{H}_k\boldsymbol{P}_k(-)\boldsymbol{H}_k^\top + \boldsymbol{H}_k\boldsymbol{M}_k + \boldsymbol{M}_k^\top\boldsymbol{H}_k^\top + \boldsymbol{R}_k\right)^{-1} \qquad (9.41)
$$

Using this optimal gain in (9.38), the updated state error covariance matrix is:

$$\begin{aligned}\boldsymbol{P}_k(+) =& \boldsymbol{P}_k(-) - \left(\boldsymbol{P}_k(-)\boldsymbol{H}_k^\top + \boldsymbol{M}_k\right)\left(\boldsymbol{H}_k\boldsymbol{P}_k(-)\boldsymbol{H}_k^\top + \boldsymbol{H}_k\boldsymbol{M}_k + \boldsymbol{M}_k^\top\boldsymbol{H}_k^\top + \boldsymbol{R}_k\right)^{-1} \\ & \left(\boldsymbol{H}_k\boldsymbol{P}_k(-) + \boldsymbol{M}_k^\top\right) \\ =& \boldsymbol{P}_k(-) - \boldsymbol{K}_k\left(\boldsymbol{M}_k^\top + \boldsymbol{H}_k\boldsymbol{P}_k(-)\right) \end{aligned} \tag{9.42}$$

Notice if $\boldsymbol{M}_k$ is zero the expression in (9.41) reduces to the standard Kalman Gain of the Extended Kalman filter (EKF) and (9.42) to the updated covariance estimate of the EKF.

9.5 Derivation of the Error State-space

In the following sections the linearizations for the EKF process and measurement model presented in sec. 3.3.1 and sec. 3.3.2 respectively, are derived. In the literature this is known as the error state-space, since the state-space is obtained with perturbations from the nomimal model and accordingly the formulated EKF is known as the Error State Kalman Filter (ESKF).

9.5.1 Preliminaries

Quaternions and rotation matrices are means for representing a rotation in the three dimensional Special Orthogonal (SO(3)) Lie group. In the following we adopt the Hamilton quaternion convention contrasted to the JPL convention [165]. A quaternion is denoted as:

$$\boldsymbol{q} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} \tag{9.43}$$

where q_w the scalar real part and $\boldsymbol{q}_v = \begin{bmatrix} q_x & q_y & q_z \end{bmatrix}^\top$ is the vector imaginary part. A rotation matrix $\boldsymbol{R}$ corresponding to a quaternion $\boldsymbol{q}$ will be denoted as $\boldsymbol{R}[q]$.

In addition, a consecutive rotation e.g. ${}^c\boldsymbol{R}_a = {}^c\boldsymbol{R}_b{}^b\boldsymbol{R}_a$ represents a rotation from the frame a to the frame b followed by a rotation to a frame c. Similarly, the same rotation is obtained with quaternions as ${}^c\boldsymbol{q}_a = {}^c\boldsymbol{q}_b \otimes {}^b\boldsymbol{q}_a$, where $\otimes$ is the quaternion multiplication defined as:

$$\boldsymbol{p} \otimes \boldsymbol{q} = \begin{pmatrix} p_w q_w - p_x q_x - p_y q_y - p_z q_z \\ p_w q_x + p_x q_w + p_y q_z - p_z q_y \\ p_w q_y - p_x q_z + p_y q_w + p_z q_x \\ p_w q_z + p_x q_y - p_y q_x + p_z q_w \end{pmatrix} \tag{9.44}$$

which using the matrix-vector product rule [166] is equivalent to:

$$\boldsymbol{p} \otimes \boldsymbol{q} = \boldsymbol{L}(\boldsymbol{p})\boldsymbol{q} = \begin{pmatrix} p_w & -\boldsymbol{p}_v^\top \\ \boldsymbol{p}_v & p_w \boldsymbol{I} + \boldsymbol{p}_{v[\times]} \end{pmatrix} \begin{pmatrix} q_w \\ \boldsymbol{q}_v \end{pmatrix} \tag{9.45}$$

$$\boldsymbol{p} \otimes \boldsymbol{q} = \boldsymbol{R}(\boldsymbol{q})\boldsymbol{p} = \begin{pmatrix} q_w & -\boldsymbol{q}_v^\top \\ \boldsymbol{q}_v & q_w \boldsymbol{I} - \boldsymbol{q}_{v[\times]} \end{pmatrix} \begin{pmatrix} p_w \\ \boldsymbol{p}_v \end{pmatrix} \tag{9.46}$$

where $[\times]$ is the skew-symmetric operation defined as:

$$\boldsymbol{q}_{v[\times]} = \begin{pmatrix} 0 & -q_z & q_y \\ q_z & 0 & -q_x \\ -q_y & q_x & 0 \end{pmatrix} \tag{9.47}$$

The exponential map is used to connect a twist $\boldsymbol{\omega}$ of the Lie algebra $so(3)$ to the $SO(3)$ Lie group:

$$e^{\boldsymbol{\omega}_{[\times]}} = \boldsymbol{I} + \boldsymbol{\omega}_{[\times]} \frac{\sin(\|\boldsymbol{\omega}\|)}{\|\boldsymbol{\omega}\|} + \boldsymbol{\omega}_{[\times]}^2 \frac{1-\cos(\|\boldsymbol{\omega}\|)}{\|\boldsymbol{\omega}\|^2} \tag{9.48}$$

which in quaternion form is:

$$e^{\boldsymbol{\omega}} = \begin{pmatrix} \cos\left(\frac{\|\boldsymbol{\omega}\|}{2}\right) \\ \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \sin\left(\frac{\|\boldsymbol{\omega}\|}{2}\right) \end{pmatrix} \tag{9.49}$$

Subsequently, assume an infinitesimal twist $\boldsymbol{\chi}$, in (9.48) and (9.49):

$$\boldsymbol{\delta q} = e^{\boldsymbol{\chi}} \approx \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\chi} \end{pmatrix} \tag{9.50}$$

$$\boldsymbol{\delta R} = e^{\boldsymbol{\chi}} \approx \boldsymbol{I} + \boldsymbol{\chi}_{[\times]} \tag{9.51}$$

This is the first-order approximation of an infinitesimal rotation, also known as the small angle approximation. To this end, if we assume a infinitesimal perturbation $\boldsymbol{\chi}$ around a nomimal rotation $\bar{\boldsymbol{R}}$, $\bar{\boldsymbol{q}}$ then:

$$\boldsymbol{R} = \bar{\boldsymbol{R}}\left(\boldsymbol{I} + \boldsymbol{\chi}_{[\times]}\right) \tag{9.52}$$

$$\boldsymbol{q} = \bar{\boldsymbol{q}} \otimes \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\chi} \end{pmatrix} \tag{9.53}$$

$$\tag{9.54}$$

In the next sections, we will formally derive the linearized error state dynamics, the proofs for the linear quantities are inspired by [166], while the derivations for the angular quantities are based on [57].

9.5.2 Relative Base Linear Velocity Dynamics

The process model for the dynamics of the base linear velocity is:

$$ {}^{b}\dot{\boldsymbol{v}}_b = -\left({}^{b}\boldsymbol{\omega}_b - \boldsymbol{b}_\omega - \boldsymbol{w}_\omega\right) \times {}^{b}\boldsymbol{v}_b + {}^{w}\boldsymbol{R}_b^\top \boldsymbol{g} + {}^{b}\boldsymbol{\alpha}_b - \boldsymbol{b}_\alpha - \boldsymbol{w}_a \tag{9.55} $$

Let a perturbation ${}^{b}\boldsymbol{\delta v}_b$ from the nomimal value ${}^{b}\bar{\boldsymbol{v}}_b$ of ${}^{b}\boldsymbol{v}_b$ so that ${}^{b}\boldsymbol{v}_b = {}^{b}\bar{\boldsymbol{v}}_b + {}^{b}\boldsymbol{\delta v}_b$. Accordingly, let $\boldsymbol{b}_\omega = \bar{\boldsymbol{b}}_\omega + \boldsymbol{\delta b}_\omega$ and $\boldsymbol{b}_\alpha = \bar{\boldsymbol{b}}_\alpha + \boldsymbol{\delta b}_\alpha$ then (9.55) becomes:

$$ {}^{b}\dot{\boldsymbol{v}}_b = {}^{b}\dot{\bar{\boldsymbol{v}}}_b + {}^{b}\boldsymbol{\delta}\dot{\boldsymbol{v}}_b \tag{9.56} $$

$$ = -\left({}^{b}\boldsymbol{\omega}_b - \bar{\boldsymbol{b}}_\omega - \boldsymbol{\delta b}_\omega - \boldsymbol{w}_\omega\right) \times \left({}^{b}\bar{\boldsymbol{v}}_b + {}^{b}\boldsymbol{\delta v}_b\right) + \left(\boldsymbol{I} - \boldsymbol{\chi}_{[x]}\right) {}^{w}\bar{\boldsymbol{R}}_b^\top \boldsymbol{g} - {}^{b}\boldsymbol{\alpha}_b - \bar{\boldsymbol{b}}_\alpha - \boldsymbol{\delta b}_\alpha - \boldsymbol{w}_\alpha \tag{9.57} $$

where in (9.57) we used the small angle approximation of ${}^{w}\boldsymbol{R}_b = {}^{w}\bar{\boldsymbol{R}}_b\left(\boldsymbol{I} + \boldsymbol{\chi}_{[\times]}\right)$. Subsequently, after eliminating products of small quantities, such as products of perturbations and noises and splitting the expression into "large signal" and "small signal" [166]:

$$ {}^{b}\dot{\bar{\boldsymbol{v}}}_b + {}^{b}\boldsymbol{\delta}\dot{\boldsymbol{v}}_b = -\left({}^{b}\boldsymbol{\omega}_b - \bar{\boldsymbol{b}}_\omega\right) \times {}^{b}\bar{\boldsymbol{v}}_b + {}^{b}\bar{\boldsymbol{v}}_b \times \left(-\boldsymbol{\delta b}_\omega - \boldsymbol{w}_\omega\right) - \left({}^{b}\boldsymbol{\omega}_b - \bar{\boldsymbol{b}}_\omega\right) \times {}^{b}\boldsymbol{\delta v}_b \tag{9.58} $$

$$ + {}^{w}\bar{\boldsymbol{R}}_b^\top \boldsymbol{g} + {}^{w}\bar{\boldsymbol{R}}_b^\top \boldsymbol{g} \times \boldsymbol{\chi} - {}^{b}\boldsymbol{\alpha}_b - \bar{\boldsymbol{b}}_\alpha - \boldsymbol{\delta b}_\alpha - \boldsymbol{w}_\alpha \tag{9.59} $$

To this end, the linearized base velocity dynamics are:

$$ {}^{b}\boldsymbol{\delta}\dot{\boldsymbol{v}}_b = {}^{b}\bar{\boldsymbol{v}}_b \times \left(-\boldsymbol{\delta b}_\omega - \boldsymbol{w}_\omega\right) - \left({}^{b}\boldsymbol{\omega}_b - \bar{\boldsymbol{b}}_\omega\right) \times {}^{b}\boldsymbol{\delta v}_b + {}^{w}\bar{\boldsymbol{R}}_b^\top \boldsymbol{g} \times \boldsymbol{\chi} - \boldsymbol{\delta b}_\alpha - \boldsymbol{w}_\alpha \tag{9.60} $$

9.5.3 Absolute Base Orientation Dynamics

The process model for the dynamics of the base orientation with respect to the world frame is:

$$ {}^{w}\dot{\boldsymbol{R}}_b = {}^{w}\boldsymbol{R}_b({}^{b}\boldsymbol{\omega}_b - \boldsymbol{b}_\omega - \boldsymbol{w}_\omega)_{[\times]} \tag{9.61} $$

which can be rewritten in quaternion form as:

$$ {}^{w}\dot{\boldsymbol{q}}_b = {}^{w}\boldsymbol{q}_b \otimes \frac{1}{2}\begin{pmatrix} 0 \\ {}^{b}\boldsymbol{\omega}_b - \boldsymbol{b}_\omega - \boldsymbol{w}_\omega \end{pmatrix} \tag{9.62} $$

where $\otimes$ denotes the quaternion multipication. In addition, for the clarity of presentation, notation with respect to frames is dropped.

Let a local perturbation $\boldsymbol{\delta q}$ from the nomimal value $\bar{\boldsymbol{q}}$ of $\boldsymbol{q}$ so that $\boldsymbol{q} = \bar{\boldsymbol{q}} \otimes \boldsymbol{\delta q}$ and $\boldsymbol{b}_\omega = \bar{\boldsymbol{b}}_\omega + \boldsymbol{\delta b}_\omega$. Subsequently, using the chain rule, the time derivative of the quaternion $\boldsymbol{q}$ is:

$$\dot{\boldsymbol{q}} = \frac{d}{dt}\left(\bar{\boldsymbol{q}} \otimes \boldsymbol{\delta q}\right) \tag{9.63}$$

$$\boldsymbol{q} \otimes \frac{1}{2}\begin{pmatrix} 0 \\ \boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega - \boldsymbol{\delta b}_\omega - \boldsymbol{w}_\omega \end{pmatrix} = \dot{\bar{\boldsymbol{q}}} \otimes \boldsymbol{\delta q} + \bar{\boldsymbol{q}} \otimes \dot{\boldsymbol{\delta q}} \tag{9.64}$$

However, the nomimal time derivative of the quaternion is:

$$\dot{\bar{\boldsymbol{q}}} = \bar{\boldsymbol{q}} \otimes \frac{1}{2}\begin{pmatrix} 0 \\ \boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega \end{pmatrix} \tag{9.65}$$

Thus, (9.64) becomes:

$$\boldsymbol{q} \otimes \frac{1}{2}\begin{pmatrix} 0 \\ \boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega - \boldsymbol{\delta b}_\omega - \boldsymbol{w}_\omega \end{pmatrix} = \bar{\boldsymbol{q}} \otimes \frac{1}{2}\begin{pmatrix} 0 \\ \boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega \end{pmatrix} \otimes \boldsymbol{\delta q} + \bar{\boldsymbol{q}} \otimes \dot{\boldsymbol{\delta q}} \tag{9.66}$$

Multiplying both sides of (9.66) with $\bar{\boldsymbol{q}}^{-1}$ and identifying that $\boldsymbol{\delta q} = \bar{\boldsymbol{q}}^{-1} \otimes \boldsymbol{q}$:

$$\boldsymbol{\delta q} \otimes \frac{1}{2}\begin{pmatrix} 0 \\ \boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega - \boldsymbol{\delta b}_\omega - \boldsymbol{w}_\omega \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 0 \\ \boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega \end{pmatrix} \otimes \boldsymbol{\delta q} + \dot{\boldsymbol{\delta q}} \tag{9.67}$$

Solving for the local perturbation $\boldsymbol{\delta \dot{q}}$:

$$\boldsymbol{\delta \dot{q}} = \boldsymbol{\delta q} \otimes \frac{1}{2}\begin{pmatrix} 0 \\ \boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega - \boldsymbol{\delta b}_\omega - \boldsymbol{w}_\omega \end{pmatrix} - \frac{1}{2}\begin{pmatrix} 0 \\ \boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega \end{pmatrix} \otimes \boldsymbol{\delta q} \tag{9.68}$$

$$= \boldsymbol{\delta q} \otimes \frac{1}{2}\begin{pmatrix} 0 \\ \boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega \end{pmatrix} + \boldsymbol{\delta q} \otimes \frac{1}{2}\begin{pmatrix} 0 \\ -\boldsymbol{\delta b}_\omega - \boldsymbol{w}_\omega \end{pmatrix} - \frac{1}{2}\begin{pmatrix} 0 \\ \boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega \end{pmatrix} \otimes \boldsymbol{\delta q} \tag{9.69}$$

where the sum of a pure quaternion is split into the sum of multiple pure quaternions. In addition, assume the small angle approximation for the quaternion $\boldsymbol{\delta q}$:

$$\boldsymbol{\delta q} = \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\chi} \end{pmatrix} \tag{9.70}$$

$$\boldsymbol{\delta \dot{q}} = \begin{pmatrix} 0 \\ \frac{1}{2}\dot{\boldsymbol{\chi}} \end{pmatrix} \tag{9.71}$$

Consequently, (9.69) takes the following form:

$$\begin{pmatrix} 0 \\ \frac{1}{2}\dot{\boldsymbol{\chi}} \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\chi} \end{pmatrix} \otimes \frac{1}{2}\begin{pmatrix} 0 \\ \boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega \end{pmatrix} + \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\chi} \end{pmatrix} \otimes \frac{1}{2}\begin{pmatrix} 0 \\ -\boldsymbol{\delta b}_\omega - \boldsymbol{w}_\omega \end{pmatrix} - \frac{1}{2}\begin{pmatrix} 0 \\ \boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega \end{pmatrix} \otimes \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\chi} \end{pmatrix} \tag{9.72}$$

Subsequently, using the matrix-vector product property of the quaternion product and defining $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega$ and $\boldsymbol{\delta}\hat{\boldsymbol{b}}_\omega = \boldsymbol{\delta b}_\omega + \boldsymbol{w}_\omega$ for simplicity:

$$\begin{pmatrix} 0 \\ \frac{1}{2}\dot{\boldsymbol{\chi}} \end{pmatrix} = \frac{1}{2}\left(\begin{bmatrix} 0 & -\hat{\boldsymbol{\omega}}^\top \\ \hat{\boldsymbol{\omega}} & -\hat{\boldsymbol{\omega}}_{[\times]} \end{bmatrix} - \begin{bmatrix} 0 & -\hat{\boldsymbol{\omega}}^\top \\ \hat{\boldsymbol{\omega}} & \hat{\boldsymbol{\omega}}_{[\times]} \end{bmatrix}\right)\begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\chi} \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & \boldsymbol{\delta}\hat{\boldsymbol{b}}^\top \\ -\boldsymbol{\delta}\hat{\boldsymbol{b}} & -\boldsymbol{\delta}\hat{\boldsymbol{b}}_{[\times]} \end{pmatrix}\begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\chi} \end{pmatrix} \tag{9.73}$$

$$= \frac{1}{2}\begin{pmatrix} 0 & \mathbf{0} \\ \mathbf{0}^\top & -2\hat{\boldsymbol{\omega}}_{[\times]} \end{pmatrix}\begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\chi} \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 \\ -\boldsymbol{\delta}\hat{\boldsymbol{b}}_\omega \end{pmatrix} \tag{9.74}$$

where in the second row of (9.74) we eliminated products of small quantities, such as products of perturbations and noises.

Finally, considering each equation individually:

$$0 = 0 \tag{9.75}$$

$$\dot{\boldsymbol{\chi}} = -\left(\boldsymbol{\omega} - \bar{\boldsymbol{b}}_\omega\right)_{[\times]}\boldsymbol{\chi} - \boldsymbol{\delta b}_\omega - \boldsymbol{w}_\omega \tag{9.76}$$

with (9.76) representing the linearized orientation dynamics and (9.75) demonstrating consistency.

9.5.4 Absolute Base Position Dynamics

The process model for the dynamics of the base position is:

$${}^{w}\dot{\boldsymbol{p}}_b = {}^{w}\boldsymbol{R}_b\,{}^{b}\boldsymbol{v}_b \tag{9.77}$$

Let the perturbation ${}^{w}\boldsymbol{\delta p}_b$ from the nomimal value ${}^{w}\bar{\boldsymbol{p}}_b$ of ${}^{w}\boldsymbol{p}_b$ so that ${}^{w}\boldsymbol{p}_b = {}^{w}\bar{\boldsymbol{p}}_b + {}^{w}\boldsymbol{\delta p}_b$, then:

$${}^{w}\dot{\bar{\boldsymbol{p}}}_b + {}^{w}\dot{\boldsymbol{\delta p}}_b = {}^{w}\bar{\boldsymbol{R}}_b\left(\boldsymbol{I} + \boldsymbol{\chi}_{[\times]}\right)\left({}^{b}\boldsymbol{v}_b + {}^{b}\boldsymbol{\delta v}_b\right) \tag{9.78}$$

$$\tag{9.79}$$

Ignoring products of perturbations and splitting the expression into ”large signal” and ”small signal”:

$$ {}^{w}\dot{\bar{\boldsymbol{p}}}_b + {}^{w}\dot{\boldsymbol{\delta p}}_b = {}^{w}\bar{\boldsymbol{R}}_b{}^{b}\boldsymbol{v}_b + {}^{w}\bar{\boldsymbol{R}}_b{}^{b}\boldsymbol{\delta v}_b - {}^{w}\bar{\boldsymbol{R}}_b{}^{b}\boldsymbol{v}_b \times \boldsymbol{\chi} \tag{9.80} $$

Thus, the linearized base position dynamics are:

$$ {}^{w}\dot{\boldsymbol{\delta p}}_b = {}^{w}\bar{\boldsymbol{R}}_b{}^{b}\boldsymbol{\delta v}_b - {}^{w}\bar{\boldsymbol{R}}_b{}^{b}\boldsymbol{v}_b \times \boldsymbol{\chi} \tag{9.81} $$

9.5.5 Absolute Support Foot Linear Velocity and IMU biases Dynamics

The process model for the dynamics of the support foot linear velocity is:

$$ {}^{w}\dot{\boldsymbol{r}}_s = \boldsymbol{w}_{r_s} \tag{9.82} $$

Let the perturbation ${}^{w}\boldsymbol{\delta r}$ from the nomimal value ${}^{w}\bar{\boldsymbol{r}}$ of ${}^{w}\boldsymbol{r}$ so that ${}^{w}\boldsymbol{r} = {}^{w}\bar{\boldsymbol{r}} + {}^{w}\boldsymbol{\delta r}$, then:

$$ {}^{w}\dot{\bar{\boldsymbol{r}}} + {}^{w}\dot{\boldsymbol{\delta r}} = \boldsymbol{w}_{r_s} \tag{9.83} $$

Nevertheless, the support foot is assumed to be stationary therefore the nomimal linear velocity is:

$$ {}^{w}\dot{\bar{\boldsymbol{r}}} = 0 \tag{9.84} $$

Therefore, the linearized suppot foot linear velocity dynamics are:

$$ {}^{w}\dot{\boldsymbol{\delta r}} = {}^{w}\boldsymbol{w}_{r_s} \tag{9.85} $$

In a similar manner, we can derive the linearized dynamics for the IMU biases:

$$ \dot{\boldsymbol{\delta b}}_\alpha = \boldsymbol{w}_\alpha \tag{9.86} $$

$$ \dot{\boldsymbol{\delta b}}_\omega = \boldsymbol{w}_\omega \tag{9.87} $$

9.5.6 Absolute Support Foot Angular Velocity Dynamics

The process model for the dynamics of the support foot angular velocity is:

$$ {}^{w}\dot{\boldsymbol{R}}_s = {}^{w}\boldsymbol{R}_s\boldsymbol{w}_{s[\times]} \tag{9.88} $$

which can be rewritten in quaternion form as:

$$ {}^{w}\dot{\boldsymbol{q}}_s = {}^{w}\boldsymbol{q}_s \otimes \frac{1}{2}\begin{pmatrix} 0 \\ \boldsymbol{w}_s \end{pmatrix} \tag{9.89} $$

For the clarity of presentation, notation with respect to frames is dropped.

Let $\boldsymbol{q} = \bar{\boldsymbol{q}} \otimes \boldsymbol{\delta q}$ so that $\dot{\boldsymbol{q}} = \dot{\bar{\boldsymbol{q}}} \otimes \boldsymbol{\delta q} + \bar{\boldsymbol{q}} \otimes \boldsymbol{\delta \dot{q}}$.

$$\dot{\bar{\boldsymbol{q}}} \otimes \boldsymbol{\delta q} + \bar{\boldsymbol{q}} \otimes \boldsymbol{\delta \dot{q}} = \boldsymbol{q} \otimes \frac{1}{2} \begin{pmatrix} 0 \\ \boldsymbol{w}_s \end{pmatrix} \tag{9.90}$$

However, the nomimal rotational velocity is:

$$\dot{\bar{\boldsymbol{q}}} = 0 \tag{9.91}$$

Therefore:

$$\bar{\boldsymbol{q}} \otimes \boldsymbol{\delta \dot{q}} = \boldsymbol{q} \otimes \frac{1}{2} \begin{pmatrix} 0 \\ \boldsymbol{w}_s \end{pmatrix} \tag{9.92}$$

Multiplying both sides from the left with $\bar{\boldsymbol{q}}^{-1}$ and using the identity $\boldsymbol{\delta q} = \bar{\boldsymbol{q}}^{-1} \otimes \boldsymbol{q}$:

$$\boldsymbol{\delta \dot{q}} = \boldsymbol{\delta q} \otimes \frac{1}{2} \begin{pmatrix} 0 \\ \boldsymbol{w}_s \end{pmatrix} \tag{9.93}$$

As before, using the matrix-vector quaternion product rule and the small angle approximation:

$$\begin{pmatrix} 0 \\ \frac{1}{2}\dot{\boldsymbol{\phi}} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & -\boldsymbol{w}_s^\top \\ \boldsymbol{w}_s & -\boldsymbol{w}_{s[\times]} \end{pmatrix} \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\phi} \end{pmatrix} \tag{9.94}$$

After eliminating the pertubration-noise products the second equation yields:

$$\dot{\boldsymbol{\phi}} = \boldsymbol{w}_s \tag{9.95}$$

while the first yields $0 = 0$ for consistency.

9.5.7 Absolute Base Position Measurement

Let the measurement be:

$${}^{w}\boldsymbol{p}_b^{\mathrm{m}} = {}^{w}\boldsymbol{p}_b + \boldsymbol{n}_{p_b} \tag{9.96}$$

Next, define the following perturbations: ${}^{w}\boldsymbol{p}_b^{\mathrm{m}} = {}^{w}\bar{\boldsymbol{p}}_b + {}^{w}\boldsymbol{\delta p}_b^{\mathrm{m}}$ and ${}^{w}\boldsymbol{p}_b = {}^{w}\bar{\boldsymbol{p}}_b + {}^{w}\boldsymbol{\delta p}_b$. Subsequently, replacing in (9.96):

$${}^{w}\bar{\boldsymbol{p}}_b + {}^{w}\boldsymbol{\delta p}_b^{\mathrm{m}} = {}^{w}\bar{\boldsymbol{p}}_b + {}^{w}\boldsymbol{\delta p}_b + \boldsymbol{n}_{p_b} \tag{9.97}$$

Therefore, the linearized absolute base position measurement is:

$$ {}^{w}\boldsymbol{\delta p}_b^{\mathrm{m}} = {}^{w}\boldsymbol{\delta p}_b + \boldsymbol{n}_{p_b} \tag{9.98} $$

9.5.8 Absolute Base Linear Velocity Measurement

Let the measurement model be:

$$ {}^{w}\boldsymbol{v}_b^{\mathrm{m}} = {}^{w}\boldsymbol{R}_b {}^{b}\boldsymbol{v}_b + \boldsymbol{n}_{v_b} \tag{9.99} $$

Again, assume the perturbations: ${}^{w}\boldsymbol{v}_b^{\mathrm{m}} = {}^{w}\bar{\boldsymbol{v}}_b + {}^{w}\boldsymbol{\delta v}_b^{\mathrm{m}}$ and ${}^{w}\boldsymbol{v}_b = {}^{w}\bar{\boldsymbol{v}}_b + {}^{w}\boldsymbol{\delta v}_b$ and use the small angle approximation for ${}^{w}\boldsymbol{R}_b$. Subsequently, (9.99) becomes:

$$ {}^{w}\bar{\boldsymbol{v}}_b + {}^{w}\boldsymbol{\delta v}_b^{\mathrm{m}} = {}^{w}\bar{\boldsymbol{R}}_b \left(\boldsymbol{I} + \boldsymbol{\chi}_{[x]}\right) \left({}^{b}\bar{\boldsymbol{v}}_b + {}^{b}\boldsymbol{\delta v}_b\right) + \boldsymbol{n}_{v_b} \tag{9.100} $$

Using the identity ${}^{w}\bar{\boldsymbol{v}}_b = {}^{w}\bar{\boldsymbol{R}}_b {}^{b}\bar{\boldsymbol{v}}_b$ and eliminating pertubation products:

$$ {}^{w}\boldsymbol{\delta v}_b^{\mathrm{m}} = {}^{w}\bar{\boldsymbol{R}}_b \boldsymbol{\chi}_{[x]} {}^{b}\bar{\boldsymbol{v}}_b + {}^{w}\bar{\boldsymbol{R}}_b {}^{b}\boldsymbol{\delta v}_b + \boldsymbol{n}_{v_b} \tag{9.101} $$

which can be rewritten as:

$$ {}^{w}\boldsymbol{\delta v}_b^{\mathrm{m}} = {}^{w}\bar{\boldsymbol{R}}_b {}^{b}\boldsymbol{\delta v}_b - {}^{w}\bar{\boldsymbol{R}}_b \left({}^{b}\bar{\boldsymbol{v}}_b \times \boldsymbol{\chi}\right) + \boldsymbol{n}_{v_b} \tag{9.102} $$

9.5.9 Absolute Base Orientation Measurement

Let the measurement model be:

$$ {}^{w}\boldsymbol{R}_b^{\mathrm{m}} = {}^{w}\boldsymbol{R}_b e^{\boldsymbol{n}_{b[\times]}} \tag{9.103} $$

which can be rewritten in quaternion form as:

$$ {}^{w}\boldsymbol{q}_b^{\mathrm{m}} = {}^{w}\boldsymbol{q}_b \otimes \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{n}_b \end{pmatrix} \tag{9.104} $$

where $\boldsymbol{n}_b$ is the quaternion from the rotation $e^{\boldsymbol{n}_b}$.
Let the perturbations, ${}^{w}\boldsymbol{q}_b^{\mathrm{m}} = {}^{w}\bar{\boldsymbol{q}}_b \otimes {}^{w}\boldsymbol{\delta q}_b^{\mathrm{m}}$ and ${}^{w}\boldsymbol{q}_b = {}^{w}\bar{\boldsymbol{q}}_b \otimes {}^{w}\boldsymbol{\delta q}_b$. Replacing in (9.104):

$$ {}^{w}\bar{\boldsymbol{q}}_b \otimes {}^{w}\boldsymbol{\delta q}_b^{\mathrm{m}} = {}^{w}\bar{\boldsymbol{q}}_b \otimes {}^{w}\boldsymbol{\delta q}_b \otimes \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{n}_b \end{pmatrix} \tag{9.105} $$

Multiplying both sides from the left with ${}^{w}\bar{\boldsymbol{q}}_b^{-1}$:

$$ {}^{w}\boldsymbol{\delta q}_b^{\mathrm{m}} = {}^{w}\boldsymbol{\delta q}_b \otimes \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{n}_b \end{pmatrix} \tag{9.106} $$

Using the matrix-product quaternion rule and the small angle approximation:

$$ \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\psi}^m \end{pmatrix} = \left(\begin{bmatrix} 1 & -\frac{1}{2}\boldsymbol{n}^\top \\ \frac{1}{2}\boldsymbol{n}_b & \boldsymbol{I} - \frac{1}{2}\boldsymbol{n}_{b[\times]} \end{bmatrix} \right) \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\chi} \end{pmatrix} \tag{9.107} $$

After eliminating noise-peturbation products, the second equation is equivalent to:

$$ \boldsymbol{\psi}^m = \boldsymbol{\chi} + \boldsymbol{n}_b \tag{9.108} $$

while the first yields $1 = 1$ for consistency.

9.5.10 Relative Support Foot Position Measurement

Let the measurement model be:

$$ {}^{b}\boldsymbol{r}_s^{\mathrm{m}} = {}^{w}\boldsymbol{R}_b^\top \left({}^{w}\boldsymbol{r}_s - {}^{w}\boldsymbol{p}_b \right) + \boldsymbol{n}_{r_s} \tag{9.109} $$

Adding the perturbations, ${}^{b}\boldsymbol{r}_s^{\mathrm{m}} = {}^{b}\bar{\boldsymbol{r}}_s + {}^{b}\boldsymbol{\delta r}_s^{\mathrm{m}}$, ${}^{w}\boldsymbol{r}_s = {}^{w}\bar{\boldsymbol{r}}_s + {}^{w}\boldsymbol{\delta r}_s$, ${}^{w}\boldsymbol{p}_b = {}^{w}\bar{\boldsymbol{p}}_b + {}^{w}\boldsymbol{\delta p}_b$, and using the small angle approximation for ${}^{w}\boldsymbol{R}_b = {}^{w}\bar{\boldsymbol{R}}_b \left(\boldsymbol{I} + \boldsymbol{\chi}_{[x]} \right)$:

$$ {}^{b}\bar{\boldsymbol{r}}_s + {}^{b}\boldsymbol{\delta r}_s^{\mathrm{m}} = \left(\boldsymbol{I} - \boldsymbol{\chi}_{[x]} \right) {}^{w}\bar{\boldsymbol{R}}_b^\top \left({}^{w}\bar{\boldsymbol{r}}_s + {}^{w}\boldsymbol{\delta r}_s - {}^{w}\bar{\boldsymbol{p}}_b - {}^{w}\boldsymbol{\delta p}_b \right) + \boldsymbol{n}_{r_s} \tag{9.110} $$

Ignoring the perturbation-noise products and using ${}^{b}\bar{\boldsymbol{r}}_s = {}^{w}\bar{\boldsymbol{R}}_b^\top \left({}^{w}\bar{\boldsymbol{r}}_s - {}^{w}\bar{\boldsymbol{p}}_b \right)$:

$$ {}^{b}\boldsymbol{\delta r}_s^{\mathrm{m}} = {}^{w}\bar{\boldsymbol{R}}_b^\top \left({}^{w}\boldsymbol{\delta r}_s - {}^{w}\boldsymbol{\delta p}_b \right) + {}^{w}\bar{\boldsymbol{R}}_b^\top \left({}^{w}\boldsymbol{r}_s - {}^{w}\boldsymbol{p}_b \right) \times \boldsymbol{\chi} + \boldsymbol{n}_{r_s} \tag{9.111} $$

9.5.11 Relative Support Foot Orientation Measurement

Let the measurement model be:

$$ {}^{b}\boldsymbol{R}_s^{\mathrm{m}} = {}^{w}\boldsymbol{R}_b^\top {}^{w}\boldsymbol{R}_s e^{n_{r[\times]}} \tag{9.112} $$

which can be rewritten in quaternion form as:

$$ {}^{b}\boldsymbol{q}_s^{\mathrm{m}} = {}^{w}\boldsymbol{q}_b^{-1} \otimes {}^{w}\boldsymbol{q}_s \otimes \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{n}_r \end{pmatrix} \tag{9.113} $$

Let ${}^b\boldsymbol{q}_s^{\mathrm{m}} = {}^b\bar{\boldsymbol{q}}_s \otimes {}^b\boldsymbol{\delta q}_s^{\mathrm{m}}$, ${}^w\boldsymbol{q}_b = {}^w\bar{\boldsymbol{q}}_b \otimes {}^w\boldsymbol{\delta q}_b$, and ${}^w\boldsymbol{q}_s = {}^w\bar{\boldsymbol{q}}_s \otimes {}^w\boldsymbol{\delta q}_s$, then:

$$ {}^b\bar{\boldsymbol{q}}_s \otimes {}^b\boldsymbol{\delta q}_s^{\mathrm{m}} = {}^{\boldsymbol{w}}\boldsymbol{\delta q}_b^{-1} \otimes {}^w\bar{\boldsymbol{q}}_b^{-1} \otimes {}^w\bar{\boldsymbol{q}}_s \otimes {}^w\boldsymbol{\delta q}_s \otimes \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{n}_r \end{pmatrix} \tag{9.114} $$

Multiplying both sides from the left with ${}^b\bar{\boldsymbol{q}}_s^{-1}$:

$$ {}^b\boldsymbol{\delta q}_s^{\mathrm{m}} = {}^b\bar{\boldsymbol{q}}_s^{-1} \otimes {}^{\boldsymbol{w}}\boldsymbol{\delta q}_b^{-1} \otimes {}^w\bar{\boldsymbol{q}}_b^{-1} \otimes {}^w\bar{\boldsymbol{q}}_s \otimes {}^w\boldsymbol{\delta q}_s \otimes \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{n}_r \end{pmatrix} \tag{9.115} $$

but ${}^b\bar{\boldsymbol{q}}_s = {}^w\bar{\boldsymbol{q}}_b^{-1} \otimes {}^w\bar{\boldsymbol{q}}_s$:

$$ {}^b\boldsymbol{\delta q}_s^{\mathrm{m}} = {}^b\bar{\boldsymbol{q}}_s^{-1} \otimes {}^{\boldsymbol{w}}\boldsymbol{\delta q}_b^{-1} \otimes {}^b\bar{\boldsymbol{q}}_s \otimes {}^w\boldsymbol{\delta q}_s \otimes \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{n}_r \end{pmatrix} \tag{9.116} $$

A triple product of quaternions can be written as:

$$ \left(\boldsymbol{q} \otimes \boldsymbol{p} \otimes \boldsymbol{q}^{-1}\right) = \begin{pmatrix} p_w \\ \boldsymbol{R}[\boldsymbol{q}]\boldsymbol{p}_v \end{pmatrix} \tag{9.117} $$

To this end:

$$ {}^b\bar{\boldsymbol{q}}_s^{-1} \otimes {}^{\boldsymbol{w}}\boldsymbol{\delta q}_b^{-1} \otimes {}^b\bar{\boldsymbol{q}}_s = \begin{pmatrix} 1 \\ -\frac{1}{2}{}^s\bar{\boldsymbol{R}}_b\boldsymbol{\chi} \end{pmatrix} \tag{9.118} $$

where the small angle approximation for ${}^w\boldsymbol{\delta q}_b^{-1} = \begin{pmatrix} 1 \\ -\frac{1}{2}\boldsymbol{\chi} \end{pmatrix}$ was used.

Furthermore, ${}^w\boldsymbol{\delta q}_s = \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\phi} \end{pmatrix}$, ${}^b\boldsymbol{\delta q}_s^{\mathrm{m}} = \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\zeta}^{\mathrm{m}} \end{pmatrix}$, and (9.116) becomes:

$$ \begin{pmatrix} 1 \\ \boldsymbol{\zeta}_s^{\mathrm{m}} \end{pmatrix} = \begin{pmatrix} 1 \\ -\frac{1}{2}{}^s\bar{\boldsymbol{R}}_b\boldsymbol{\chi} \end{pmatrix} \otimes \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\phi} \end{pmatrix} \otimes \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{n}_r \end{pmatrix} \tag{9.119} $$

Using the right matrix-product quaternion rule:

$$ \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\zeta}_s^{\mathrm{m}} \end{pmatrix} = \begin{pmatrix} 1 & -\frac{1}{2}\boldsymbol{\phi}^\top \\ \frac{1}{2}\boldsymbol{\phi} & \boldsymbol{I} - \frac{1}{2}\boldsymbol{\phi}_{[x]} \end{pmatrix} \begin{pmatrix} 1 \\ -\frac{1}{2}{}^s\bar{\boldsymbol{R}}_b\boldsymbol{\chi} \end{pmatrix} \otimes \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{n}_r \end{pmatrix} \tag{9.120} $$

After eliminating products of pertubation-noise:

$$ \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\zeta}_s^{\mathrm{m}} \end{pmatrix} = \begin{pmatrix} 1 \\ -\frac{1}{2}{}^s\bar{\boldsymbol{R}}_b\boldsymbol{\chi} + \frac{1}{2}\boldsymbol{\phi} \end{pmatrix} \otimes \begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{n}_r \end{pmatrix} \tag{9.121} $$

Using the right matrix-product quaternion rule one more time:

$$\begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\zeta}_s^{\mathrm{m}} \end{pmatrix} = \begin{pmatrix} 1 & -\frac{1}{2}\boldsymbol{n}_r^\top \\ \frac{1}{2}\boldsymbol{n}_r & \boldsymbol{I} - \frac{1}{2}\boldsymbol{n}_r \end{pmatrix} \begin{pmatrix} 1 \\ -\frac{1}{2}{}^s\bar{\boldsymbol{R}}_b\boldsymbol{\chi} + \frac{1}{2}\boldsymbol{\phi} \end{pmatrix} \tag{9.122}$$

After products of pertubation-noise are eliminated, (9.122) is equivalent to:

$$\begin{pmatrix} 1 \\ \frac{1}{2}\boldsymbol{\zeta}_s^{\mathrm{m}} \end{pmatrix} = \begin{pmatrix} 1 \\ -\frac{1}{2}{}^s\bar{\boldsymbol{R}}_b\boldsymbol{\chi} + \frac{1}{2}\boldsymbol{\phi} + \frac{1}{2}\boldsymbol{n}_r \end{pmatrix} \tag{9.123}$$

To this end, the second equation yields:

$$\boldsymbol{\zeta}_s^{\mathrm{m}} = -{}^w\bar{\boldsymbol{R}}_s^\top {}^w\bar{\boldsymbol{R}}_b\boldsymbol{\chi} + \boldsymbol{\phi} + \boldsymbol{n}_r \tag{9.124}$$

while the first one yields $1 = 1$ for consistency.

9.6 Derivation of Closed-Form Outlier Detection

In this section the closed-form expression for the outlier measurement detection algorithm, presented in section 4.2.3, is derived.

$$\begin{aligned} \boldsymbol{B}_k &= \int (\boldsymbol{y}_k^o - \boldsymbol{h}^o(\boldsymbol{x}_k))(\boldsymbol{y}_k^o - \boldsymbol{h}^o(\boldsymbol{x}_k))^\top p(\hat{\boldsymbol{x}}_k) d\boldsymbol{x}_k \\ &= \boldsymbol{y}_k^o \boldsymbol{y}_k^{o\top} - 2\boldsymbol{y}_k^o \int \boldsymbol{h}^o(\boldsymbol{x}_k)^\top p(\hat{\boldsymbol{x}}_k) d\boldsymbol{x}_k \\ &+ \int \boldsymbol{h}^o(\boldsymbol{x}_k)\boldsymbol{h}^o(\boldsymbol{x}_k)^\top p(\hat{\boldsymbol{x}}_k) d\boldsymbol{x}_k \end{aligned} \tag{9.125}$$

using the first order approximation of (4.8) post to the update:

$$\boldsymbol{h}^o(\boldsymbol{x}) = \boldsymbol{h}(\hat{\boldsymbol{x}}_k^+) + \boldsymbol{H}_k^o(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_k^+)$$

the first integral of (9.125) can be computed as:

$$\begin{aligned} \int \boldsymbol{h}^o(\boldsymbol{x}_k)^\top p(\hat{\boldsymbol{x}}_k) d\boldsymbol{x}_k &= \int \boldsymbol{h}^o(\hat{\boldsymbol{x}}_k^+)^\top p(\hat{\boldsymbol{x}}_k) d\boldsymbol{x}_k \\ + \int (\boldsymbol{x}_t - \hat{\boldsymbol{x}}_k^+)^\top \boldsymbol{H}_k^{o\top} p(\hat{\boldsymbol{x}}_k) d\boldsymbol{x}_k &= \boldsymbol{h}^o(\hat{\boldsymbol{x}}_k^+)^\top \end{aligned}$$

while the second integral of (9.125) is equivalent to:

$$\int \boldsymbol{h}^o(\boldsymbol{x}_k)\boldsymbol{h}^o(\boldsymbol{x}_k)^\top p(\hat{\boldsymbol{x}}_k)d\boldsymbol{x}_k = \int (\hat{\boldsymbol{x}}_k^+ + \boldsymbol{H}_k^o(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^+))$$
$$(\hat{\boldsymbol{x}}_k^+ + \boldsymbol{H}(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_k^+))^\top p(\hat{\boldsymbol{x}}_k)d\boldsymbol{x}_k = \hat{\boldsymbol{x}}_k^+\hat{\boldsymbol{x}}_k^{+\top} + \boldsymbol{H}_k^o\boldsymbol{P}_k^+\boldsymbol{H}_k^{o\top}$$

Thus (9.125) becomes:

$$\begin{aligned}\boldsymbol{B}_k &= \boldsymbol{y}_k^o\boldsymbol{y}_k^{o\top} - 2\boldsymbol{y}_k^o\boldsymbol{h}^o(\hat{\boldsymbol{x}}_k^+)^\top \\ &\quad + \boldsymbol{h}^o(\hat{\boldsymbol{x}}_k^+)\boldsymbol{h}^o(\hat{\boldsymbol{x}}_k^+)^\top + \boldsymbol{H}_k^o\boldsymbol{P}_k^+\boldsymbol{H}_k^{o\top}\end{aligned} \tag{9.126}$$

List of Acronyms

DoF Degree of Freedom

DRC Darpa Robotic Challenge

ZMP Zero Moment Point

COP Center of Pressure

CoM Center of Mass

IMU Inertial Measurement Unit

F/T Force/Torque

FRI Foot Rotation Indicator

CWC Contact Wrench Cone

GSN Gait Sensitivity Norm

GRF Ground Reaction Force

LIPM Linear Inverted Pendulum Model

IPM Inverted Pendulum Model

FIP Floating-based Inverted Pendulum

KF Kalman Filter

LKF Linear Kalman Filter

EKF Extended Kalman Filter

UKF Unscented Kalman Filter

QP Quadratic Program

CP Capture Point

DCM Divergent Component of Motion

MPC Model Predictive Control

PFL Partial Feedback Linearization

LTI Linear Time Invariant

LTV Linear Time Varying

HZD Hybrid Zero Dynamics

PHZD Partial Hybrid Zero Dynamics

ROS Robot Operating System

ME Momentum Estimator

ESKF Error State Kalman Filter

RGESKF Robust Gaussian Error State Kalman Filter

VO Visual Odometry

LO LIDAR Odometry

SS Single Support

DS Double Support

LSS Left Single Support

RSS Right Single Support

SEROW State Estimation RObot Walking

GEM Gait-Phase Estimation Module

KFusion Kinect Fusion

SVO Semi-dense Visual Odometry

SLAM Simultaneous Localization and Mapping

PCA Principal Component Analysis

NN Neural Network

GMM Gaussian Mixture Model

EM Expectation Maximization