KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Generalizing Human-Demonstrated Robot Skills

Using Invariant Models for Manipulation Tasks in Contact

Ali Mousavi Mohammadi

Supervisor:
Prof. dr. ir. Herman Bruyninckx
Em. prof. dr. ir. Joris De Schutter
Dr. ir. Erwin Aertbeliën

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Mechanical Engineering

June 2025

Generalizing Human-Demonstrated Robot Skills

Using Invariant Models for Manipulation Tasks in Contact

Ali MOUSAVI MOHAMMADI

Examination committee:
Em. prof. dr. ir. Bob Puers, chair
Prof. dr. ir. Herman Bruyninckx, supervisor
Em. prof. dr. ir. Joris De Schutter, co-supervisor
Dr. ir. Erwin Aertbeliën, co-supervisor
Dr. ir. Wilm Decré
Dr. Gianni Borghesan
Prof. dr. ir. Tinne De Laet
Prof. Dr. Oliver Brock
(Technische Universität Berlin)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Mechanical Engineering

June 2025

© 2025 KU Leuven – Faculty of Engineering Science
Uitgegeven in eigen beheer, Ali Mousavi Mohammadi, Celestijnenlaan 300 box 2402, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

Use of Generative AI: The generative AI assistance tools were not used during the research/writing process of this thesis, except for mere language assistance. The text/code/images in this thesis are author's own (unless otherwise specified) and generative AI has only been used in accordance with the KU Leuven guidelines and appropriate references have been added. The content has been reviewed and edited as needed and the author takes full responsibility for the content of the thesis.

Preface

This thesis marks the culmination of several years of research, learning, and personal growth. I would like to take this opportunity to express my sincere gratitude to those who have supported and accompanied me throughout this journey.

First and foremost, I am deeply grateful to my two main advisors, Professor Joris De Schutter and Dr Erwin Aertbeliën, for their outstanding guidance, encouragement, and trust. Their knowledge, insight, and unwavering support have been vital in shaping both the direction and the quality of my research. I would also like to thank Professor Herman Bruyninckx, who kindly accepted to formally join my advisory team toward the end of this project, ensuring a smooth administrative process following Professor De Schutter's retirement.

I am also thankful to my colleagues and collaborators. I choose not to mention names here, simply because I fear unintentionally leaving someone out. Working with such talented and generous individuals has made this journey not only possible but also enjoyable and inspiring. In particular, I would like to express my heartfelt appreciation to the colleagues in the ERC team, with whom I shared many stimulating meetings and wonderful moments. Your energy and friendship made a big difference.

A heartfelt thank you goes to my wife and daughter, whose love, patience, and constant encouragement gave me the strength to persevere, even during the most demanding times. Their presence has been a true source of joy and balance in my life.

Finally, I would like to thank my parents, who laid the foundation for my education and personal values. Their sacrifices, encouragement, and unconditional support have been with me every step of the way.

To all of you, I am profoundly grateful.

Abstract

In robotics, contact tasks involve manipulating objects while in contact with the environment or other objects, and they are prevalent across various fields, including industrial automation, domestic robotics, and healthcare. In this thesis, we focus on the manipulation of a single object attached to a robot end effector in contact with another object in the environment. These tasks encompass a wide range of applications, such as assembly, surface following, and interacting with articulated objects like doors or drawers. Programming such tasks is particularly challenging because it requires precise coordination and control of both the object's motion and its interaction forces with the environment.

Learning from Demonstration (LfD) has the potential to simplify the programming of contact-rich tasks by making it more intuitive, allowing non-expert users to teach robots without requiring in-depth knowledge of robotics. However, despite these advantages, LfD still faces several challenges. Generalizing learned behaviors to new, unseen situations remains difficult, as robots need to adapt to varying conditions without explicit programming. Furthermore, data collection is expensive and data efficiency is a key issue, requiring the system to learn effectively from a small number of examples. Additionally, ensuring accurate calibration between multiple sensors is crucial for achieving reliable performance. These challenges highlight the need for robust methods to improve the practicality and reliability of LfD in real-world robotic applications.

The objective of this thesis is to provide invariant models that are efficient to build, easy to fine tune, and applicable to real-world robot tasks in contact. These task models are derived from both motion and interaction force data that are collected during human demonstrations of the task and that are represented in a new form, making it easier to learn and generalize the task by a robot.

The first contribution of this work is the introduction of invariant descriptors for

force trajectories in contact tasks, building on the duality between motion and force. This provides a novel representation of force trajectories that complements existing descriptors for motion trajectories. The work also offers robust methods and software for calculating these descriptors, even in the presence of noisy measurements, using optimal control. Additionally, the approach is experimentally validated using recorded data from human demonstrations of two contact tasks, yielding task representations that are invariant to reference frame calibration and sensor placement.

The second contribution is the development of an automatic method for deriving the optimal reference frame, or optimal task frame, directly from motion and force data recorded during task demonstrations. This method eliminates the need for expert-defined frames or predefined candidates by using screw theory to generate frame candidates and a probabilistic approach to determine the optimal one. It is robust to uncertainties in geometric features and works across various types of motion and force tasks, enabling invariant, accurate, and versatile task representation. The method has been validated through experiments showcasing its effectiveness in supporting learning-based controller design for contact tasks.

The third contribution is the introduction of a fast and inexpensive closed-form method for estimating the force/torque sensor offsets, as well as the gravity parameters, i.e. mass and center of mass of the rigid object mounted on the force/torque sensor. This method leverages screw theory and relies solely on force and moment measurements, hence without requiring accurate motion data. The approach has been validated through both simulations and practical experiments, demonstrating its effectiveness in real-world applications, such as gravity compensation for force control of contact tasks.

The fourth and final contribution is the derivation of relations between invariant descriptors and their corresponding moving frames, each derived from data expressed with respect to one of the two interacting rigid bodies. Although these relations were originally developed at the beginning of the PhD program and provided valuable insights, they are presented as the final chapter of this thesis. This placement allows for a deeper reflection on the derived relations and their implications, while also enabling the integration of data and results from the earlier chapters related to the first and second contributions, offering a more comprehensive understanding of the topic.

Beknopte samenvatting

In de robotica omvatten contacttaken het manipuleren van objecten terwijl deze in contact zijn met de omgeving of andere objecten. Deze taken komen veel voor in diverse domeinen, zoals industriële automatisering, huishoudrobots en de gezondheidszorg. In deze thesis richten we ons op de manipulatie van een enkel object dat is bevestigd aan de eind-effector van een robot en in contact staat met een ander object in de omgeving. Dergelijke taken omvatten een breed scala aan toepassingen, zoals assemblage, het volgen van oppervlakken, en het omgaan met gearticuleerde objecten zoals deuren of lades. Het programmeren van zulke taken is bijzonder uitdagend, omdat het een precieze coördinatie en controle vereist van zowel de beweging van het object als de interactiekrachten met de omgeving.

Learning from Demonstration (LfD), oftewel leren uit demonstraties, heeft het potentieel om het programmeren van taken met veel contact aanzienlijk te vereenvoudigen. Het maakt het programmeren intuïtiever en stelt niet-experten in staat om robottaken aan te leren zonder diepgaande kennis van robotica. Ondanks deze voordelen blijft LfD kampen met enkele belangrijke uitdagingen. Het generaliseren van aangeleerd gedrag naar nieuwe, onbekende situaties is moeilijk, omdat robots zich moeten kunnen aanpassen aan variërende omstandigheden zonder expliciete programmering. Bovendien is het verzamelen van data kostbaar en is data-efficiëntie cruciaal: het systeem moet effectief kunnen leren uit een beperkt aantal voorbeelden. Tot slot is een nauwkeurige kalibratie tussen verschillende sensoren essentieel voor betrouwbare prestaties. Deze uitdagingen onderstrepen de noodzaak voor robuuste methoden om de praktische toepasbaarheid en betrouwbaarheid van LfD in de echte wereld te verbeteren.

Het doel van deze thesis is het ontwikkelen van invariante modellen die efficiënt op te bouwen zijn, gemakkelijk af te regelen, en toepasbaar op taken in de echte wereld waarbij de robot in contact treedt met zijn omgeving. Deze taakmodellen worden opgesteld op basis van bewegings- en contactkrachtdata die verzameld

worden tijdens de demonstratie van de taak door een menselijke operator en die voorgesteld worden in een nieuwe vorm waardoor het eenvoudiger wordt voor een robot om een taak te leren en te generaliseren.

De eerste bijdrage van dit werk is de introductie van invariante descriptoren voor krachttrajecten in contacttaken, gebaseerd op de dualiteit tussen beweging en kracht. Dit levert een nieuwe representatie op van krachttrajecten die complementair is aan bestaande descriptoren voor bewegingstrajecten. Het werk bevat tevens robuuste methoden en software om deze descriptoren te berekenen, zelfs in aanwezigheid van meetruis, gebruikmakend van optimale controle. De methode is experimenteel gevalideerd met opgenomen data van menselijke demonstraties van twee contacttaken, en levert taakrepresentaties op die invariant zijn ten opzichte van referentiekalibratie en sensorplaatsing.

De tweede bijdrage is de ontwikkeling van een automatische methode om het optimale referentieframe, oftewel het optimale taakframe, rechtstreeks af te leiden uit bewegings- en krachtdata verzameld tijdens taakdemonstraties. Deze methode elimineert de noodzaak voor expertgedefinieerde frames of vooraf bepaalde kandidaten, door gebruik te maken van schroeftheorie om frame-kandidaten te genereren en een probabilistische benadering om het optimale frame te bepalen. Ze is robuust ten opzichte van onzekerheden in geometrische kenmerken en werkt voor diverse soorten bewegings- en krachttaken, waardoor een invariante, nauwkeurige en veelzijdige taakrepresentatie mogelijk wordt. De methode is gevalideerd via experimenten die haar effectiviteit aantonen bij het ondersteunen van het ontwerp van de aansturing voor het uitvoeren van aangeleerde taken in contact.

De derde bijdrage is de introductie van een snelle en goedkope rekenmethode in gesloten vorm voor het schatten van offsetwaarden van kracht-/momentsensoren, alsook de zwaartekrachtparameters, zoals de massa en het massamiddelpunt van het starre object dat op de sensor is gemonteerd. Deze methode maakt gebruik van schroeftheorie en steunt uitsluitend op kracht- en momentmetingen, dus zonder nauwkeurige bewegingsdata nodig te hebben. De methode is gevalideerd via zowel simulaties als praktische experimenten, en heeft haar nut bewezen in toepassingen zoals zwaartekrachtcompensatie voor krachtgestuurde contacttaken.

De vierde en laatste bijdrage is de afleiding van verbanden tussen invariante descriptoren en hun overeenkomstige bewegende referentieframes, elk afgeleid uit gegevens die zijn uitgedrukt ten opzichte van één van de twee interagerende starre lichamen. Hoewel deze verbanden oorspronkelijk werden ontwikkeld aan het begin van het doctoraatsonderzoek en waardevolle inzichten opleverden, worden ze opgenomen in het slothoofdstuk van deze thesis. Deze plaatsing laat toe om dieper te reflecteren over de afgeleide verbanden en hun implicaties,

en maakt ook een integratie mogelijk van data en resultaten uit de eerdere hoofdstukken die betrekking hebben op de eerste en tweede bijdrage. Zo ontstaat een meer omvattend begrip van het onderwerp.

List of Abbreviations

ASA Average Srew Axis. 170, 171

ASIP Average Screw-axes Intersection Point. 86, 87, 90–93, 104, 118, 132, 133, 141, 144, 145, 170, 172

AVOF Average Vector Orientation Frame. 84–87, 92, 94, 95, 118, 170

COM Center of Mass. xxii, xxiii, 17, 125–135, 138, 140, 142, 144–146, 166, 170, 172

DOF Degree of Freedom. 47, 55, 57, 59, 62, 69, 88, 100–102, 105, 111, 181

EU European Union. 1

FS Frenet-Serret. 31–35, 37, 39–41, 55, 148–151, 156, 161, 173, 175, 176, 183, 187, 191, 198, 199

FT force/torque. xxii, 3, 15, 17, 124–130, 135, 139, 141, 142, 144–146, 166

HRI Human Robot Interaction. 21

ICR Instantaneous Center of Rotation. 11

ISA Instantaneous Screw Axis. 12, 13, 15, 39–42, 55–58, 66, 148, 149, 151–153, 160, 161, 170, 171, 183, 189, 191, 192, 198, 199, 203, 208, 210, 214, 215, 217, 219, 220

LfD Learning from Demonstration. 9, 16, 72–78, 164

MSE Mean-Squared Error. 35, 37, 46

OCP Optimal Control Problem. 6, 35–37, 44, 45, 48, 49, 52, 55, 57, 60, 61, 63, 65, 68, 161, 164, 170, 171, 206, 213, 215

RMS Root Mean Square. 136, 138, 142, 144, 145

RMSE Root Mean Square Error. 112, 116, 117, 171

SVD Singualr Value Decomposition. 84–86

TCP Tool Center Point. 48, 51, 57–59, 62

List of Symbols

$\mathcal{A}$	rigid body named *"$\mathcal{A}$"* corresponds to viewpoint *"$\mathcal{A}$"* .
a	reference point named *"a"* attached to a rigid body or the world.
$\{a\}$	reference pose frame named *"a"* attached to a rigid body or the world.
$[a]$	reference orientation frame named *"a"* attached to a rigid body or the world.
${}^{b}_{c}\boldsymbol{p}_{a}$	3×1 position vector representing the position of point *"b"* with respect to point *"a"* expressed in reference frame $[c]$. (If the context is clear, indices might be omitted for simplicity. If *"a"* and *"c"* are the same, *"c"* is omitted for simplicity.)
${}^{b}_{c}\boldsymbol{e}_{i\,a}$	3×1 unit vector representing the ith column of a rotation matrix ${}^{b}_{c}\boldsymbol{R}_{a}$. (If the context is clear, indices might be omitted for simplicity. If *"a"* and *"c"* are the same, *"c"* is omitted for simplicity.)
${}^{b}_{c}\boldsymbol{R}_{a}$	3×3 rotation matrix representing the orientation of frame $[b]$ with respect to frame $[a]$ expressed in reference frame $[c]$. (If the context is clear, indices might be omitted for simplicity. If *"a"* and *"c"* are the same, *"c"* is omitted for simplicity.)
${}^{b}_{c}\boldsymbol{T}_{a}$	4×4 homogeneous transformation matrix representing the orientation and position of frame $\{b\}$ with respect to frame $\{a\}$ and expressed in frame $[c]$. (If the context is clear, indices might be omitted for simplicity. If *"a"* and *"c"* are the same, *"c"* is omitted for simplicity.)
${}^{b}_{c}\boldsymbol{c}^{d}_{a}$	3×1 general vector representing a quantity associated with rigid body *"b"* with respect to rigid body *"a"* at reference point d and expressed in reference orientation frame $[c]$. (If the context is clear, indices might be omitted for simplicity.)
${}^{b}_{c}\boldsymbol{\omega}_{a}$	3×1 rotational velocity vector of rigid body *"b"* with respect to rigid body *"a"* expressed in reference orientation frame $[c]$. (If the context is clear, indices might be omitted for simplicity.)
${}^{b}_{c}\boldsymbol{v}^{d}_{a}$	3×1 translational velocity vector of rigid body *"b"* with respect to rigid body *"a"* at reference point d and expressed in reference orientation frame $[c]$. (If the context is clear, indices might be omitted for simplicity.)

${}^{b}_{c}\boldsymbol{f}_{a}$	3×1 force vector acting on the rigid body *"a"* exerted by *"b"* and expressed in reference orientation frame $[c]$. (If the context is clear, indices might be omitted for simplicity.)
${}^{b}_{c}\boldsymbol{m}^{d}_{a}$	3×1 moment vector acting on the rigid body *"a"* exerted by *"b"* at reference point d and expressed in reference orientation frame $[c]$. (If the context is clear, indices might be omitted for simplicity.)
${}^{b}_{c}\boldsymbol{\mathfrak{z}}^{d}_{a}$	6×1 general screw representing a quantity associated with rigid body *"b"* with respect to rigid body *"a"* at reference point d and expressed in reference orientation frame $[c]$. This screw consists of two 3×1 vectors known as directional component ${}^{b}_{c}\boldsymbol{a}_{a}$ and moment component ${}^{b}_{c}\boldsymbol{b}^{d}_{a}$. (If the context is clear, indices might be omitted for simplicity.)
${}^{b}_{c}\boldsymbol{t}^{d}_{a}$	6×1 screw twist of rigid body *"b"* with respect to rigid body *"a"*. The first three elements are the rotational velocity vector ${}^{b}_{c}\boldsymbol{\omega}_{a}$ and the last three elements are the translational velocity vector ${}^{b}_{c}\boldsymbol{v}^{d}_{a}$. (If the context is clear, indices might be omitted for simplicity.)
${}^{b}_{c}\boldsymbol{w}^{d}_{a}$	6×1 screw wrench acting on the rigid body *"a"* exerted by rigid body *"b"*. The first three elements are the force vector ${}^{b}_{c}\boldsymbol{f}_{a}$ and the last three elements are the moment vector ${}^{b}_{c}\boldsymbol{m}^{d}_{a}$. (If the context is clear, indices might be omitted for simplicity.)
${}^{b}\boldsymbol{S}_{a}$	6×6 screw transformation matrix that, when multiplied with a screw twist or wrench, transforms the screw from $\{b\}$ to $\{a\}$.
ξ	geometric progress variable.
ω_{κ}	local curvature of a point trajectory.
ω_{τ}	local torsion of a point trajectory.
$\boldsymbol{i}$	3×1 vector representing moving frame invariants of a vector trajectory, consisting of ω_{κ} and ω_{τ}.
$\widetilde{\boldsymbol{c}}$	3×1 vector representing object invariant of a vector trajectory, consisting of c.
$\widetilde{\boldsymbol{R}}$	3×3 rotation matrix representing the FS frame.
v_t	translational velocity of the origin of the ISA frame along $\boldsymbol{e}_t$.
v_b	translational velocity of the origin of the ISA frame along $\boldsymbol{e}_b$.
$\boldsymbol{\dot{\imath}}$	6×1 screw representing moving frame invariants of a screw trajectory, consisting of ω_{κ}, ω_{τ}, v_t, and v_b.
$\widetilde{\boldsymbol{\mathfrak{z}}}$	6×1 screw representing object invariants of a screw trajectory, consisting of a and b.
$\widetilde{\boldsymbol{T}}$	4×4 homogeneous transformation matrix representing the ISA frame.
ϵ	desired tolerance on the trajectory reconstruction cost over the entire horizon in the OCP formulation.
L	desired length scale in the OCP formulation to properly compare rotational and translational invariants.

$\breve{\boldsymbol{\jmath}}$	6×1 screw with the average subtracted.
$\boldsymbol{C}$	uncentered covariance matrix.
σ_i^2	ith singular value achieved through a singular value decomposition.
$\hat{\boldsymbol{C}}_w$	6×6 compliance matrix.
$\hat{C}_f$	scalar value representing the translational component of compliance.
$\hat{C}_m$	scalar value representing the rotational component of compliance.
k_p	feedback gain of a position constraint.
k_w	feedback gain of a force constraint.
w_p	weight assigned to a position constraint.
w_w	weight assigned to a force constraint.
k_{eq}	equivalent stiffness.
b_{eq}	equivalent damping.
$\{w\}$	world pose frame.
$\{b\}$	body pose frame.
$\{o_i\}$	ith object pose frame.
$\{tcp\}$	tool center point pose frame.
$\{fs\}$	force/torque sensor pose frame.
$\{tr\}$	tracker pose frame.
$\{tf\}$	task pose frame.
$\{tl\}$	tool pose frame.
$[mf]$	moving orientation frame in which invariants of a vector trajectory are defined.
$[rot]$	moving orientation frame in which invariants of rotation trajectory are defined.
$[tra]$	moving orientation frame in which invariants of translation trajectory are defined.
$[for]$	moving orientation frame in which invariants of force trajectory are defined.
$[mom]$	moving orientation frame in which invariants of moment trajectory are defined.
$\{mf\}$	moving pose frame in which invariants of a screw trajectory are defined.
$\{isam\}$	moving pose frame in which invariants of motion trajectory are defined.
$\{isaw\}$	moving pose frame in which invariants of wrench trajectory are defined.

Contents

Abstract iii

Beknopte samenvatting v

List of Abbreviations x

List of Symbols xi

Contents xv

List of Figures xxi

List of Tables xxv

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Research Objectives . . . 6
- 1.3 Background and Related Work . . . 7
  - 1.3.1 Task Frame . . . 7
  - 1.3.2 Learning from Demonstration . . . 9
  - 1.3.3 Screw Theory . . . 11
  - 1.3.4 Invariant Models . . . 14
- 1.4 Contributions . . . 15
- 1.5 Outline . . . 17

2 Invariant Descriptors of Motion and Force Trajectories 19
- 2.1 Introduction . . . 21
  - 2.1.1 Invariant Trajectory Descriptors for Motion . . . 23
  - 2.1.2 Objective and Contributions . . . 25
- 2.2 Preliminaries . . . 26
  - 2.2.1 Rigid-body Kinematics . . . 27

2.2.2 Duality between Twist and Wrench as General Screws . 29
2.2.3 Time-invariance Using a Geometric Progress Variable . 29
2.3 Invariant Descriptor for Vector Trajectories 30
2.3.1 Definition of Invariants 31
2.3.2 Robust Calculation of Invariants Using Optimal Control 34
2.4 Invariant Descriptor for Screw Trajectories 38
2.4.1 Definition of Invariants 39
2.4.2 Robust Calculation of Invariants Using Optimal Control 42
2.5 Application to 3D Contour Following 46
2.5.1 Experimental Set-up 47
2.5.2 Objectives 48
2.5.3 Experiment Design 49
2.5.4 Data Processing 51
2.5.5 Results and Discussion 53
2.6 Application to Peg-on-Hole Alignment 61
2.6.1 Experimental Set-up and Data Processing 62
2.6.2 Results and Discussion 63
2.7 Discussion and Conclusion 66

3 Automatic Derivation of an Optimal Task Frame for Contact-Rich Tasks 71
3.1 Introduction 72
3.1.1 Problem and Research Objective 73
3.1.2 Approach 74
3.1.3 Contribution and Outline 75
3.2 Relation to the state of the art 75
3.2.1 Collection of Demonstration Data 75
3.2.2 Assumptions and Applications 76
3.2.3 Selection of Task Frame 77
3.2.4 Task Model and Control 78
3.2.5 Generalization and Skill Transfer 79
3.3 Preliminaries and Notation 80
3.3.1 Orientation and Pose 80
3.3.2 Velocity Kinematics 82
3.3.3 Wrench and General Screw 82
3.3.4 Time-invariant Motion and Wrench Trajectories 83
3.3.5 Average Vector Orientation Frame (AVOF) 84
3.3.6 Average Screw-axes Intersection Point (ASIP) 86
3.4 Derivation of the Task Frame 88
3.4.1 Input Data and Expected Outcome 88
3.4.2 Conceptual Approach 89
3.4.3 Selection of Origin (Figure 3.6a) 92
3.4.4 Selection of Orientation (Figure 3.6b) 94

3.4.5 Representation of Data in the Task Frame (Figure 3.6c) 97
3.5 Experiments Design . . . 99
3.5.1 Applications . . . 99
3.5.2 Demonstration Setup . . . 103
3.5.3 Pre-Processing of Demonstration Data . . . 103
3.5.4 Task Frame Derivation and Data Post-Processing . . . 104
3.5.5 Task Performance Setup and Control Strategy . . . 104
3.5.6 Scenarios . . . 107
3.6 Experimental Results . . . 108
3.6.1 Task Frame Derivation . . . 108
3.6.2 Control Performance . . . 112
3.7 Discussion and Conclusion . . . 118

4 Estimating Sensor Offsets and Gravity Parameters Using Wrench Measurements 123
4.1 Introduction . . . 125
4.2 Preliminaries . . . 128
4.3 Estimating the Gravity Parameters and Sensor Offsets . . . 128
4.3.1 Assumptions . . . 128
4.3.2 Baseline: Least-Squares Method . . . 130
4.3.3 Proposed Three-Step Estimation Method . . . 131
4.4 Simulation Results . . . 134
4.5 Experimental Validation and Results . . . 139
4.5.1 Experimental Setup . . . 139
4.5.2 Experimental Methodology . . . 140
4.5.3 Experimental Results . . . 142
4.6 Conclusions . . . 145

5 Relations Between Invariant Descriptors and Moving Frames in Two Viewpoints 147
5.1 Introduction . . . 148
5.2 Relations between Moving Frames . . . 149
5.2.1 FS Frames . . . 149
5.2.2 ISA Frames . . . 151
5.3 Relations between Invariants . . . 153
5.3.1 Vector Invariants . . . 154
5.3.2 Screw Invariants . . . 157
5.4 Conclusion . . . 161

6 Conclusion 163
6.1 Contributions . . . 163
6.2 Limitations . . . 167
6.3 Future Work . . . 170

A Visualization of Old and New Notations of FS Frames **173**

B Relation between the Kinematics of the FS Frame and the FS Equation **175**

C Combining Orientation Estimates **177**

D Equivalent Stiffness and Damping **181**

E Relative Orientation of FS Frames Derived from Data Expressed in Two Viewpoints **183**
E.1 Vectors and their Derivatives in Two Viewpoints 183
E.2 Unit Vectors of FS Frames in Two Viewpoints 184
E.2.1 Tangent Axis of FS Frames 184
E.2.2 Normal Axis of FS Frames 185
E.2.3 Binormal Axis of FS Frames 186
E.3 Rotation Matrix between Two FS Frames 187
E.4 Special Case: No Rotation between Rigid Bodies 187
E.5 Special Case: Relative Orientation of FS Frames of Rotation . 187

F Relative Position of ISA Frames Derived from Data Expressed in Two Viewpoints **189**
F.1 Screws and their Derivatives in Two Viewpoints 189
F.2 Position of ISA Frames in Two Viewpoints 191
F.2.1 Position of the First ISA Frame in the First Viewpoint . 192
F.2.2 Position of the Second ISA Frame in the Second Viewpoint 194
F.3 Transformation Matrix between Two ISA Frames 198
F.4 Special Case: No Rotation and/or No Translation between Rigid Bodies 198
F.4.1 No Rotation between Rigid Bodies 198
F.4.2 No Rotation and No Translation between Rigid Bodies . 199
F.5 Special Case: Relative Position of ISA Frames of Motion 199

G Validation of Relations between Invariant Descriptors **205**
G.1 Validation through Synthetic Data 205
G.1.1 Vector Invariants 205
G.1.2 Screw Invariants 206
G.2 Validation through Real Experimental Data 213

H Discretized Formulas for Calculation of Screw Trajectory **217**

I Discretized Formulas for Calculation of Screw Invariants **219**

J Generalization Across New Objects **223**

Bibliography **225**

List of Publications **237**

List of Figures

1.1 Population pyramids in 2019 and 2050 2
1.2 Different examples of contact tasks investigated in the literature 3
1.3 Different examples of generalization investigated in the literature 5
1.4 Real-world applications of contour following task 7
1.5 Specification of Door Opening task 8
1.6 Providing data for LfD methods 10
1.7 The screw theory applied to the 2-D rigid body motion 12
1.8 The screw theory applied to the 3-D rigid body motion 13
1.9 Two examples of screw-theory-based models in the literature . 14
1.10 Thesis outline 18

2.1 Visualization of screw twist and screw wrench in the world frame and in the body frame 28
2.2 Invariant descriptor for a vector trajectory 32
2.3 Invariant descriptor for a screw trajectory 40
2.4 The 3D contour following setup 47
2.5 The six different configurations of the motion tracker on the tool 50
2.6 Screw and vector invariants calculated from position and orientation of tracker or tool point on manipulated tool, measured in the world frame 54
2.7 Screw and vector invariants calculated from measured force and moment trajectories, transformed to either the tracker or the tool center frame 56
2.8 Screw invariants calculated from measured wrench transformed to the world frame 58
2.9 Screw invariants calculated from a simulated wrench in the world frame to study the reduction in variation with respect to Figure 2.8 58
2.10 Moving frames calculated from data of Trial 5 60

2.11 Reconstruction of the motion and wrench trajectory of Trial 7 at a new location given the screw invariants of the measured trajectory seen from the world 61
2.12 Peg-on-hole setup with defined frames and illustration of the contact 62
2.13 Screw invariants for motion and force for the peg-on-hole task . 64
2.14 Moving frames for trials in the peg-on-hole alignment task, all seen from the world frame 65

3.1 The typical workflow versus the proposed approach for robot task learning 73
3.2 The similarity transformation under general conditions with three moving frames 81
3.3 Average vector orientation frame 85
3.4 Average Screw-axes intersection point 87
3.5 Four models with a perfect decoupling between the direction and moment components of a screw 90
3.6 Flow chart of the proposed approach 94
3.7 Role of alignment and calculation of weighted average of two orientation frames 96
3.8 Setups used for teaching different tasks to the robot 100
3.9 Schematic representation of the proposed constraint-based control strategy 107
3.10 2-D Contour Following performed by the UR10e robot 114
3.11 Difference in orientation of the tool in the Drawing task 115
3.12 Peg on Hole Alignment performed by the UR10e robot 116
3.13 Bottle Opening performed by the UR10e 116
3.14 Expert task frame (pale) and derived task frame (bright) for the Drawing task 118

4.1 An operator demonstrates a manipulation task involving contact, where they use a bottle opener to remove the cap from a bottle. 126
4.2 Visualization of the configuration of the FT sensor and the object mounted on it. The COM frame, denoted as $\{com\}$, is parallel to the FT sensor frame $\{ft\}$ and is located at the object's COM. The gravity force is always aligned with the vertical axis of the world frame $\{w\}$. 129
4.3 Overview of the proposed method. This includes three main steps: (1) estimation of mass and force offsets, (2) estimation of COM, and (3) estimation of moment offsets. 134
4.4 Range of excitation shown by Euler angles in XYZ format . . . 135

4.5 Setups used for the experimental validation. (a) The object and the Schunk FT sensor are mounted on the end-effector of the UR10e robot arm. (b) The object and the JR3 FT sensor as well as a motion tracker are mounted on a demonstration tool to be held by a human operator. 139
4.6 Experimental setup for analyzing the estimation of mass, COM, and sensor offsets with the robot. The object is rotated with a specified angle (here: $\pm 70°$) around two perpendicular axes, the X and Y axes of the FT sensor frame. 140
4.7 Experimental setup for analyzing the estimation of gravity parameters and sensor offsets within the human demonstration context. The operator is asked to rotate the object similarly to the motion of the robot's end-effector. 141

5.1 Scene graph at orientation level for frames 150
5.2 FS frames calculated in two viewpoints 151
5.3 Scene graph at pose level for frames 152
5.4 ISA frames calculated in two viewpoints 153
5.5 Scene graph at velocity level for orientation frames 155
5.6 Scene graph at twist level for frames 158

A.1 Visualization of the XYZ frame versus the TNB frame 173

G.1 Ground truth screw invariants of motion 206
G.2 Ground truth screw invariants of wrench 206
G.3 Relative motion of the two rigid bodies 207
G.4 Relative twist between the two rigid bodies 207
G.5 Interaction wrench between the two rigid bodies 208
G.6 Screw invariants of motion calculated from twist data expressed in two viewpoints 209
G.7 Relation between the invariants of motion 209
G.8 Rotational (upper row) and translational distance (lower row) between the two ISA frames of motion 210
G.9 Investigation of loop closure described in (5.23) for motion analysis 211
G.10 Screw invariants of wrench calculated from wrench data expressed in two viewpoints 211
G.11 Distance between the two ISA frames of wrench calculated by (G.2) from wrench data expressed in two viewpoints 212
G.12 Investigation of loop closure described in (5.23) for wrench analysis 212
G.13 The angle and distance between the two moving frames are calculated in different ways 213
G.14 Screw invariants of motion derived from pose data expressed in two viewpoints 214

G.15 Screw invariants of wrench derived from wrench data expressed in two viewpoints 214
G.16 The distance between the two ISA frames of motion is calculated by (G.2) 215
G.17 The distance between the two ISA frames of wrench is calculated by (G.2) 215

I.1 Two consecutive ISAs and the common normal between them . 220

J.1 Task frames and their adapted versions for new objects 224

List of Tables

2.1 Overview of trajectory types of which invariant descriptors can be calculated 25
2.2 Overview of notation used throughout the chapter 26
2.3 Overview of equations for vector invariants and screw invariants 43
2.4 Twelve demonstration trials is listed with one of the six configurations of the motion tracker and with the artificial transformation that was applied to the measured poses 51
2.5 Reconstruction error for the motion and force trajectory 61

3.1 Vectors of interest for determination of orientation and geometric progress variable 92
3.2 Specifications of the force/torque sensors 103
3.3 Control parameters used in the experiments 107
3.4 Outcome of the proposed approach for selection of the task frame 110
3.5 Robot control performance 113

4.1 Monte Carlo simulation for estimating the gravity parameters and sensor offsets using the baseline and proposed methods. . . 137
4.2 Specifications of the force/torque sensors 140
4.3 Estimating the gravity parameters and sensor offsets using the baseline and proposed methods. 143

5.1 Name of different FS frames according to the type of the vector 149
5.2 Name of different ISA frames according to the type of the screw 152
5.3 Vector invariants with complete notation in two viewpoints . . 154
5.4 Screw invariants with complete notation in two viewpoints . . . 157

Chapter 1

Introduction

1.1 Motivation

Industrial robots were initially introduced to perform repetitive, tedious, dangerous, or large-scale production tasks, typically in environments isolated from humans. This separation was essential to prevent accidents and ensure the safety of both humans and robots because robots had no communication with their environment [44]. In these structured environments, robots were carefully programmed beforehand, minimizing the risk of errors and damage [49].

On the other hand, according to [2], the proportion of employed individuals aged 55 and over in the European Union (EU) increased from 12% in 2004 to 20% in 2019. This trend reflects the aging of the workforce and underscores the growing demographic challenges faced by the EU. Looking ahead, projections indicate that these challenges will intensify in 2050, shown in Figure 1.1. The EU's working-age population is expected to decline. Simultaneously, the proportion of individuals aged 65 and over is projected to rise. These demographic shifts suggest that labor shortages will become an increasingly significant challenge in the coming decades. As the working-age population contracts and the proportion of older individuals grows, there will be heightened pressure on labor markets and social support systems.

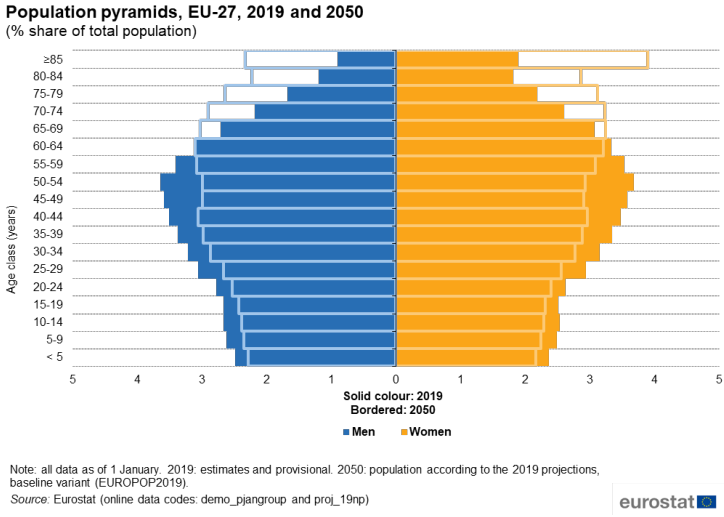
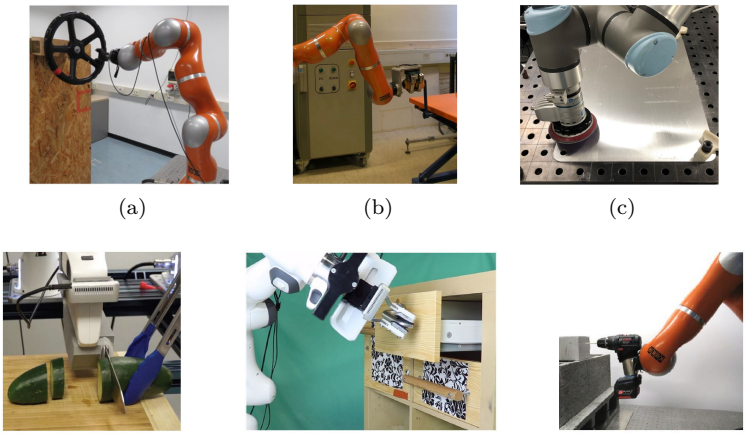


Figure 1.1: Population pyramids in 2019 (the solid bars) and 2050 (the bars with borders). Image from [1].

One solution to these issues is technological innovation. Robots have become essential in addressing labor shortages caused by aging populations and the growing need for efficiency. Therefore, the expectations for robots have grown significantly as means to mitigate the labor shortage. Advances in manufacturing technologies, computational power, vision systems, sensors, and actuators have enabled robots to assume greater responsibility in the workforce by performing more complex tasks, moving beyond preprogrammed operations. This progress has driven the need for robots that can adapt to unstructured, varying environments and handle variability and uncertainty in tasks, products, and environments. These evolving expectations introduce several key challenges. First, robots must be capable of handling variability and uncertainty, not only in the physical characteristics of the products they manipulate but also in task conditions and environmental factors. Second, they must support the rapid deployment of new task variations within a product family or class, without requiring extensive reprogramming or manual intervention, an especially important capability in low-volume production settings, where traditional reprogramming is both time-consuming and costly. Lastly, robots are increasingly expected to operate in shared workspaces and adapt to human interaction, necessitating more robust, flexible, and intuitive behavior.

Despite rising expectations and these advancements, programming robots for

complex tasks remains a challenge. For certain tasks, particularly those involving rich physical contact, explicit programming is impossible for non-experts and even difficult for experts. In contact-rich tasks, physical interactions between the robot and its environment impose specific geometric constraints, which must be carefully accounted for by the robot controller. Examples of these tasks, illustrated in Figure 1.2, highlight the necessity of precise force control. Typically, contact force is measured with a wrist force/torque (FT) sensor and the control policy must accurately account for this force. Failure to do so can result in damage to both the robot and its surroundings, making accurate and adaptive control crucial.



(a) (b) (c)

(d) (e) (f)

Figure 1.2: Different examples of contact tasks investigated in the literature. (a) Image from [8] (b) Image from [39] (c) Image from [56] (d) Image from [58] (e) Image from [64] (f) Image from [80].

To meet the growing demands outlined above, researchers have turned to advanced learning paradigms, such as deep learning, reinforcement learning, and learning from human demonstrations. These approaches aim to equip robots with the ability to acquire skills autonomously and adapt them to novel situations. Importantly, robots must learn from humans to take over repetitive and non-ergonomic tasks while maintaining the flexibility needed for such activities. This allows humans to concentrate on precise and dexterous tasks that demand even greater flexibility [75] and/or a complex sequence of skills. This collaboration between robots and humans ensures an efficient division of labor. Such capabilities are particularly valuable in contact-rich tasks, where conventional programming struggles to address the complexities of physical

interactions and the variability of real-world scenarios. By leveraging these learning paradigms, robots can better generalize their skills to new objects and unforeseen environments, as shown in Figure 1.3. Ultimately, robots must be designed to be easily programmable, enabling them to adapt more effectively to new products and unstructured and varying environments, thereby meeting the needs of modern, flexible production lines. This transition to adaptive, generalizable, learning-based systems represents a critical step in enabling robots to fulfill the high expectations placed on them in varying, unstructured settings.

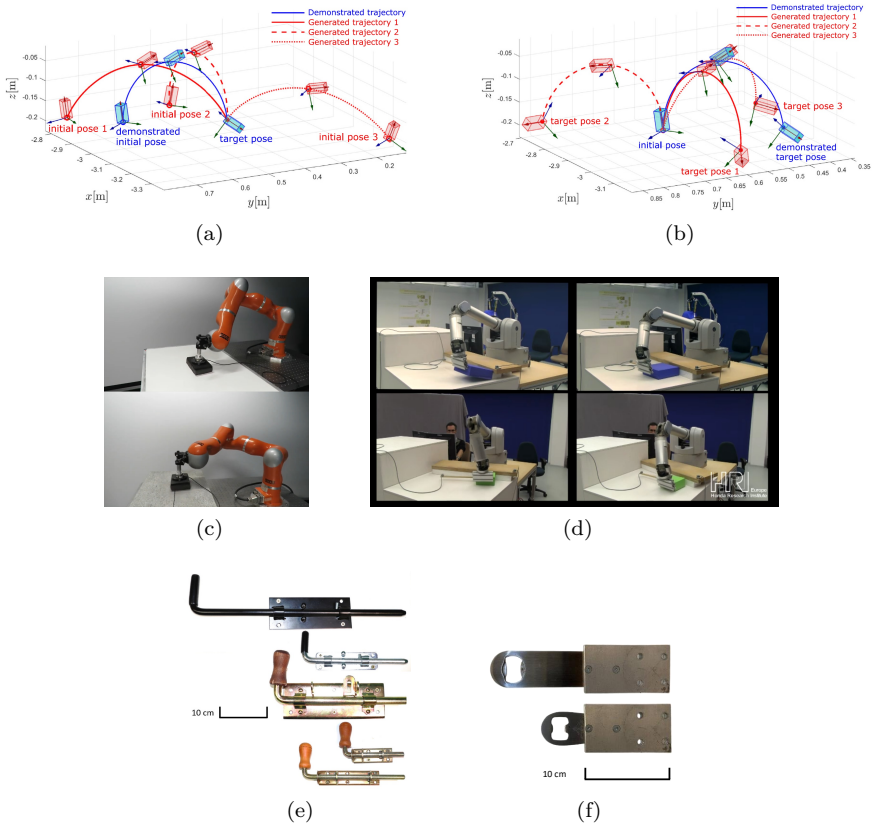


Figure 1.3: Different examples of generalization investigated in the literature. (a) The pick-and-place skill generalizes to novel initial positions and orientations [107]. (b) The same task generalizes to novel target positions and orientations [107]. (c) The surface-wiping task generalizes to a new surface (concrete) different from the one used in the demonstration (board) [80]. (d) The pull-and-flip skill generalizes to new positions and orientations, different initial object placements, and various object sizes [57]. (e) The lock-opening task generalizes to different locks compared to the one used in the demonstration [64]. (f) The bottle-opening skill generalizes to a new tool, specifically a bottle opener that is longer than the one used in the demonstration [103].

1.2 Research Objectives

The overall objective of this research is to develop an approach for modeling manipulation tasks in contact from a limited number of human demonstrations. This approach needs to have an invariant representation to ensure generalizability to unseen scenarios. Specifically, the model must be independent of the chosen reference frames, whether on the object of interest (tool) or the world, to guarantee invariance. This independence not only enhances the model's generalizability across different scenarios but also reduces the reliance on extensive setup calibration, simplifying its deployment. Additionally, the model must capture only the essential information of the demonstrated task, further improving its generalizability.

The method proposed in [34] introduces an invariant representation of motion using analytical expressions involving higher-order derivatives. While theoretically elegant, this method has the disadvantage that it suffers from sensitivity to measurement noise due to its reliance on these derivatives. Moreover, it is prone to singularities, where small disturbances in the data can lead to large amplification of errors. To mitigate these limitations, the problem is reformulated as an Optimal Control Problem (OCP) in [106], which provides a more robust approach to handling noise and singularities. However, tuning the OCP for new tasks remains challenging, and the optimization process is susceptible to convergence to local optima, requiring careful initialization. These issues might be the main reasons for the limited adoption of invariant methods by other researchers. Although the interest in these methods has increased recently, it is only limited to the case of motion.

A more recent approach proposed in [10] addresses these issues by introducing a motion representation with more global behavior, which reduces sensitivity to noise and singularities and eliminates the need for extensive tuning of the OCP.

On the other hand, building on the foundation laid by [34, 106] which introduced an invariant representation for motion, this research extends the concept to include force data, focusing on advancing towards real-world, contact-rich applications. Consequently, the model needs to integrate both motion and force data to effectively capture the complexities of contact tasks. Additionally, this work aims to develop a global invariant representation of contact tasks based on both motion and force, inspired by the method proposed in [10].

This research also aims to address a wide range of tasks, characterized by different predominant modes of motion and interaction: predominant translation (e.g., opening a drawer), predominant rotation (e.g., opening a door), predominant force (e.g., writing), or predominant torque (e.g., unscrewing a bolt), or any combination of these. The proposed model is expected to handle both articulated

objects (e.g., doors, drawers) and non-articulated objects (e.g., surface-following and contour-following tasks), providing a unified approach across a variety of task types. While some tasks, such as contour following, discussed in Chapters 2 and 3, may appear simplistic or illustrative, they reflect the core challenges found in real-world applications such as surface finishing [56] and cheese decrusting [6], illustrated in Figures 1.4a and 1.4b, respectively.



(a) (b)

Figure 1.4: Two real-world applications of contour following tasks: (a) surface finishing by a sander mounted on the robot end-effector [56], (b) cheese decrusting by a sharp spinning blade mounted on the robot end-effector [6].

Furthermore, given the goal of enabling practical applications of manipulation tasks in contact, the model must demonstrate the ability to generalize to novel scenarios and new objects.

To achieve these objectives, this thesis proposes two types of invariant representations for contact tasks. These representations aim to contribute toward making robotic systems more adaptable and robust in real-world applications. Ultimately, these developments are intended to address two challenges in industrial robotics: handling variability and uncertainty in tasks, products, or environments, and enabling rapid deployment of new task variations within a product family or class, especially where traditional automation is not cost-effective, such as in low-volume production series.

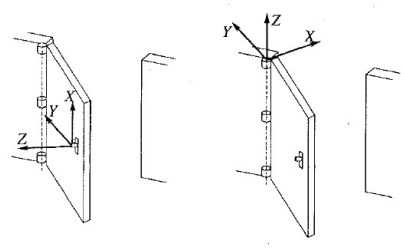
1.3 Background and Related Work

1.3.1 Task Frame

To respect the geometric constraints imposed by the environment, a carefully designed mechanical compliance can be added to the system to make it possible

to manage the uncertainties. This is typically known as passive compliance. In addition to passive compliance, which is fixed and cannot handle large uncertainties, it is possible to design a programmable compliance within the force control scheme that can manage large uncertainties based on force feedback. The specification of a contact task and its controller are expressed in a reference frame, commonly known as task frame in the context of robot force control. The concept of task frame was introduced in [71, 86] and was formalized in [23]. A set point for either desired force or desired velocity can be specified along and around each axis of the task frame. Then, the low-level controller is responsible to follow them.

The task frame is defined according to the expert user's knowledge. For example, in the task shown in Figure 1.5, an expert may choose to program the task using a frame attached either to the door knob or to the hinge. The task specifications required for programming differ accordingly: those associated with the knob-attached frame are listed on the left, while those for the hinge-attached frame are shown on the right. Each choice offers specific advantages and reflects a different perspective: either pulling the door through its knob or rotating the door around its hinge. Using a knob-attached frame requires less modeling effort, as it does not account for the door's rotation about the hinge. In contrast, the hinge-attached frame can result in lower control errors by aligning more closely with the actual motion of the door. Ultimately, the selection of the task frame depends on the expert's objective and modeling preferences.



move compliantly {
with task frame directions
xt: force 0 N
yt: force 0 N
zt: velocity v mm/sec
axt: force 0 Nmm
ayt: force 0 Nmm
azt: force 0 Nmm
} until distance > d mm

move compliantly {
with task frame directions
xt: force 0 N
yt: force 0 N
zt: force 0 N
axt: force 0 Nmm
ayt: force 0 Nmm
azt: velocity -ω rad/sec
} until time > t sec

Figure 1.5: Specification of Door Opening task by force and velocity set points, proposed in two frames: one attached to the door knob and the other one attached to the door's hinge. Image from [23].

The concept of task frame could specify constraints according to the desired set points. More advanced constrained-based methods have been developed based upon the concept of task frame such as [35, 9]. Although originally inspired by the task frame concept, the approach in [9] is much more general because the one proposed in [35] requires a kinematic loop that includes the feature frames (and accordingly feature coordinates) to specify complex tasks of general sensor-based robot systems. In [9], the approach uses more general feature variables to specify complex sensor-based tasks. These feature variables can be related to more than one task frame. These constraints can be specified in terms of position, velocity, or force.

1.3.2 Learning from Demonstration

This field has been referred to by various names, including Learning from Demonstration (LfD), Programming by Demonstration, and Imitation Learning. Despite the differences in terminology, all of these refer to methods aimed at extracting a mathematical model from data recorded during a human demonstration of a specific task. Inspired by the natural imitation abilities of humans and animals [28], these methods enable non-experts to teach robots new tasks with minimal or no programming.

Typically, these methods involve four main steps:

- Data collection: The task is demonstrated multiple times, and the corresponding data is recorded. This means each demonstration has several trials, as shown in Figure 1.6a.
- Data segmentation: The recorded data is divided into segments, each corresponding to an individual trial, as illustrated in Figure 1.6b.
- Trial alignment: The segmented trials are aligned either geometrically or through methods such as dynamic time warping, as shown in Figure 1.6c.
- Model extraction: A model is derived from the segmented and aligned trials.

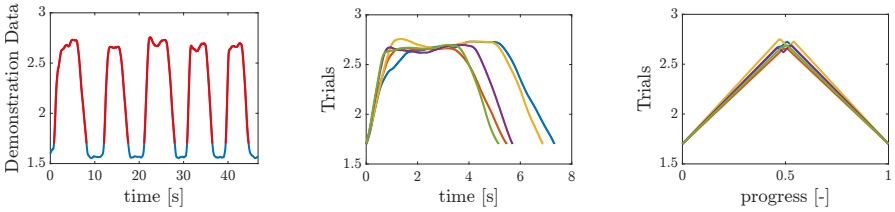


Figure 1.6: Providing data for LfD methods includes the following steps: (a) multiple demonstrations of a task in blue and segmented trials in red according to a threshold, (b) segmented data as individual trials, and (c) alignment of the trials using a geometric progress.

The number of trials required depends on the applied method and the task: some methods can operate effectively with just a single demonstration, also known as one-shot learning [64, 56, 63]; others may require quite a few demonstrations to achieve effective results [103, 119, 100]. Using as few trials as possible, can significantly expedite the task programming process. However, this approach may also make it more challenging to extract a representative model from the demonstrated data.

Typically, as long as the task required of a robot closely resembles the demonstrations, most methods perform effectively. This is analogous to interpolation within the range of the demonstrated data. However, challenges emerge in unseen situations, such as operating in new coordinates or with novel tools. These scenarios involve extrapolation beyond the demonstrated data and are often referred to as generalization in the learning community. A robust model should be capable of handling unseen situations effectively, requiring minimal or no adjustments. To achieve this, an approach is needed to distinguish between essential and non-essential (contextual) information. Subsequently, a model must be extracted from the essential information, preserving the intrinsic features of the task that are relevant to the application.

This can also be viewed from a constraint-based perspective. The approach should effectively identify essential constraints from the demonstrated data, with each constraint representing a consistent property or relation. A specific set of constraints can define a skill and be maintained over an interval, known as a segment.

1.3.3 Screw Theory

Screw theory provides a framework for describing the relative motion of two rigid bodies and the wrenches acting between them using specific axes. In this context, a wrench refers to the combined representation of force and moment. The motion of a rigid body is represented by an axis, which combines linear and angular velocity, while the wrench interaction is characterized by another axis, combining force and moment. Simplified versions of screw theory for 2-D rigid body kinematics and the wrench concept for 3-D rigid body statics are commonly introduced in undergraduate mechanical engineering courses. In the following two paragraphs, screw theory is briefly introduced in the context of rigid body motion; the application of screw theory to wrench interaction is discussed in detail in the next chapter.

The concept is shown in Figure 1.7a for rigid body motions in 2-D where the axis is seen as a point from this viewpoint. In a general case which considers two rigid bodies specified by two reference frames, body $\{b\}$ and world $\{w\}$, a general 2-D motion of the body can be shown as a rotation around an *instantaneous center of rotation* (ICR). The two special cases in 2-D are illustrated in Figures 1.7b and 1.7c, corresponding to purely translational and purely rotational motions, respectively. Pure rotation refers to the motion of the rigid body around a fixed point, with no translational component. In an ideal scenario with noise-free data, the ICR lies at infinity (i.e., $r = \infty$) in the case of pure translation. This can be interpreted as rotating on a circle with an infinite radius. The ICR lies at a fixed finite location in the case of pure rotation (i.e., $r = cte$). However, with real-world data where noise is inevitable, the ICR for the translational case will still be located far from the rigid body (i.e., with a large r), while in the rotational case, the ICR will remain relatively stable and close to a fixed point (i.e., $r \approx cte$)

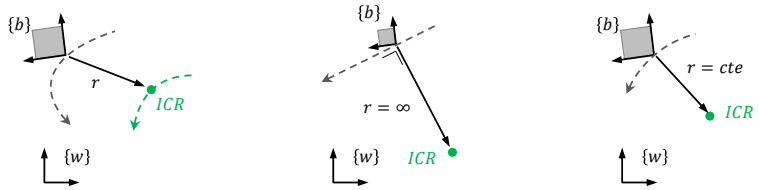


Figure 1.7: The screw theory applied to the 2-D rigid body motion. (a) The Instantaneous Center of Rotation (ICR) depicted for a general case with a moving ICR and variable r. (b) The ICR is depicted for translation, the first special case, which results in moving ICR to infinity $r = \infty$. (c) The ICR is depicted for rotating around a point, the second special case, which results in a fixed ICR, $r = cte$.

This concept can be extended to a 3-D problem as shown in Figure 1.8a where now the *instantaneous screw axis* (ISA) is visible. Apart from the general case, there are two special cases for 3-D problems similar to the 2-D motions. The first case is shown in Figure 1.8b, where there is no rotation between rigid bodies, meaning the rigid body is purely translating. Thus, the axis cannot be defined (i.e., lies at infinity $r = \infty$). The second case occurs when one rigid body rotates around a fixed line. In this case, the axis remains parallel to itself as shown in Figure 1.8c (i.e., $r = cte$). As with the 2-D cases, real-world data with noise affects the ISA. In the case of pure translation, the estimated screw axis appears far from the rigid body (i.e., with a large r), while in pure rotation, the axis remains at a relatively stable and consistent distance (i.e., $r \approx cte$).

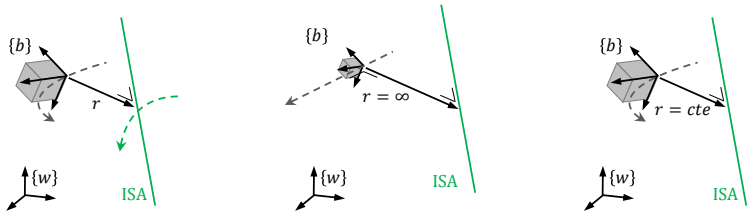


Figure 1.8: The screw theory applied to the 3-D rigid body motion. (a) The Instantaneous Screw Axis (ISA) depicted for a general case with a moving ISA and variable r. (b) The ISA is depicted for translation, the first special case, which results in moving ISA to infinity $r \to \infty$. (c) The ISA is depicted for rotation, the second special case, which results in an almost fixed ISA $r \approx cte$.

In [81], screw theory is integrated with an extensive affordance framework to enable remote, contact-rich tasks with a mobile manipulator. By leveraging screw theory, the affordance framework significantly improves teleoperation efficiency, achieving reductions in task duration and applied forces compared to direct teleoperation. A proposed approach in [61] implicitly maps and satisfies geometric constraints inherent in a human demonstration, enabling the generation of motion trajectories. It leverages the natural parameterization of screw representation to ensure that the generated trajectories respect the task's underlying geometric features. The work in [69] combines screw theory with kinesthetic demonstration to extract coordinate invariant representation of the manipulation constraints from the demonstrated data. The idea is to segment a task by a sequence of constant screw motions and use the segmented screws to generate motion where it does not rely on the reference frame. In [14], a novel screw-based framework is proposed for teaching robots bimanual manipulation tasks by learning from human video demonstrations and refining skills through interaction. Their proposed method incorporates three types of screw models and selects the most suitable one for the demonstrated bimanual task by comparing a score function that quantifies the distance between the demonstrated trajectory and each model. It is worth noting that, in the literature, screw-based motion models are often derived for specific cases. Two such examples are illustrated in Figure 1.9a and Figure 1.9b. These models are typically defined based on the screw pitch, denoted by h, which represents the ratio of translation along the ISA to the rotation about the ISA.



(a) (b)

Figure 1.9: Two examples of screw-theory-based models in the literature. (a) In [14], three types of motions are considered: revolute, revolute3D (meaning revolute with fixed object orientation, right hand in this figure), and prismatic. (b) In [81], four types of motions are considered: fixed, pure rotation ($h = 0$), pure translation ($h = \infty$), both rotation and translation ($0 < h < \infty$).

1.3.4 Invariant Models

Invariance has been shown to offer key advantages across various robotics domains, addressing several limitations of conventional, coordinate-dependent methods. In control, invariant formulations ensure consistent closed-loop behavior regardless of motion direction [113], and help overcome issues of dimensional inconsistency and unit dependence in hybrid force/position control by relying on reciprocal rather than orthogonal relations [40, 66, 67, 36, 26]. In estimation, invariant approaches such as the Invariant Extended Kalman Filter achieve better convergence by expressing correction terms on Lie groups [16], with successful applications in attitude estimation for UAVs [22]. In motion generation, invariant properties such as equi-affine curvature and shape preservation allow consistent adaptation and deformation of trajectories regardless of the reference frame [18, 72, 82, 17, 61, 107]. In action recognition, invariant descriptors and metrics enable reliable recognition across variations in viewpoint, position, occlusion, and scale [27, 115]. By leveraging these invariant properties, we can construct more generalizable, compact, and robust task models, particularly valuable in human-robot interaction scenarios where demonstrations are limited and environmental variability is high.

This thesis focuses on developing task models that are invariant to the location and orientation of reference frames. To achieve this, we use invariant descriptors, rooted in screw theory, as our chosen invariant model. The invariants express

not only the relative motion and interaction wrench between the rigid bodies but also the motion of the ISA. This is done within a frame defined on the ISA. The invariants remain inherently unchanged under changes of body or world reference frames and, with minor modifications, can also exhibit invariance to factors such as scale and time. This representation is introduced for rigid body motion trajectories [34] and point motion trajectories [104].

While invariant descriptors explain how the ISA is moving instantaneously, a novel representation is proposed in [10] that can express the motion information in an average reference frame. This frame is associated with a time series of screw axes that represents the average screw axis and average screw axis intersection point. Extending invariant models proposed in [34, 104, 10] to wrench trajectories could be particularly beneficial for contact tasks, where a skill is defined by a set of constraints that remain valid within a given segment.

Models that are designed to be invariant to the choice of reference frames offer a significant advantage, as they minimize the need for extensive calibration of motion and FT sensors. Such invariance is crucial because relying on fixed sensor configurations reduces the generalizability of a model and increases its sensitivity to calibration errors. Without this property, a learned model is typically valid only within a narrowly defined experimental setup, making it less suitable for real-world applications where precise calibration may not always be feasible or available.

In addition to reference frame invariance, adaptability to changes in geometry is another desirable property in task modeling. These models can accommodate variations in task conditions, such as changes in tool or environment through the adjustment of a limited number of model parameters. This is particularly important when applying a model to different instances within the same object category. For example, in a Door Opening scenario illustrated in Figure 1.5, an effective model should not only define a task frame that is invariant to changes in world or body reference frames, but also support modifications that account for variations such as the distance between the hinge and the door knob. This combination of reference frame invariance and parameter-based adaptability enables the transfer and reuse of task models across related but distinct scenarios, improving both robustness and scalability.

1.4 Contributions

This section provides a summary of the four main contributions of this thesis.

1. A Novel Method for Modeling Contact Tasks Using Invariant Descriptors

Invariant descriptors for motion trajectories were introduced in [34] while the invariants are introduced for force trajectories in this thesis for the first time. This is developed for screw and vector trajectories through a novel optimization-based approach.

2. A Novel Method for Modeling and Generalizing Contact-Rich Tasks Using Screw Theory and Constraint-Based Control Strategy

Although the task frame was introduced in the early 80s, its selection or design remains dependent on the expert user. Location and orientation of the task frame must be chosen by the expert as explained by [86] as follows.

> *Though selection of the constraint frame's position and orientation is a matter of discretion, a suitable choice can make the task of determining the natural constraints particularly simple.*

However, the orientation is addressed in [23] as follows,

> *the TF directions should provide maximal decoupling.*

and the location is addressed as follows which are both more accurately defined compared to [86].

> *The most optimal location of the TF is the one in which the TF directions are maximally decoupled.*

But this guideline has not been investigated mathematically or discussed in depth.

Through the third chapter of this thesis, a framework is proposed to derive the task frame from data recorded during human demonstration. This can be considered as an answer with mathematical support to the issue of selection of location and orientation of the task frame. Then, this automatically-derived task frame is used in the context of LfD to validate the proposed approach where a constraint-based controller is developed to perform the learned tasks.

3. A Novel Method for Estimating FT Sensor Offsets and Gravity Parameters Using only Wrench Measurements

Estimating the FT sensor offsets and gravity parameters, i.e. mass and center of mass (COM), using only the data coming from the FT sensor is the third novelty of this thesis. There is no need for orientation measurements nor carefully designed orientations. This contribution directly benefits from the methods developed in the previous contribution.

4. A thorough Discussion on the Invariant Descriptors Derived from Different Viewpoints

As the name of invariant descriptors claims, they are invariant with respect to some changes. But what if we calculate them with respect to different viewpoints, i.e. calculate them from data expressed with respect to different rigid bodies in contact. The last contribution of this thesis is about comparing the descriptors derived from different viewpoints that perform a relative motion with respect to each other. Thus, the mathematical relations between the invariants and their frames are explained in detail.

1.5 Outline

This thesis provides approaches to build invariant models for segmented contact-rich tasks from two perspectives: instantaneously (i.e. locally) or as a whole (i.e. globally). Figure 1.10 provides a schematic overview of the thesis outline, described in detail below where Chapters 2 and 5 study local approaches while Chapters 3 and 4 deals with global approaches.

- **Chapter 2** presents a novel optimization-based approach to model the contact tasks using invariant descriptors.
- **Chapter 3** presents a novel method to derive an optimal reference frame from demonstration data to model, control, and generalize contact tasks.
- **Chapter 4** can be regarded as a focused exploration derived from the methodological approach developed in the previous chapter, showcasing a specific application: estimating the FT sensor offsets and gravity parameters of rigid bodies.
- **Chapter 5** provides an extensive discussion on the relations between the invariant models proposed with respect to different viewpoints.

- **Chapter 6** provides conclusions and discussion of the methods proposed in this thesis.

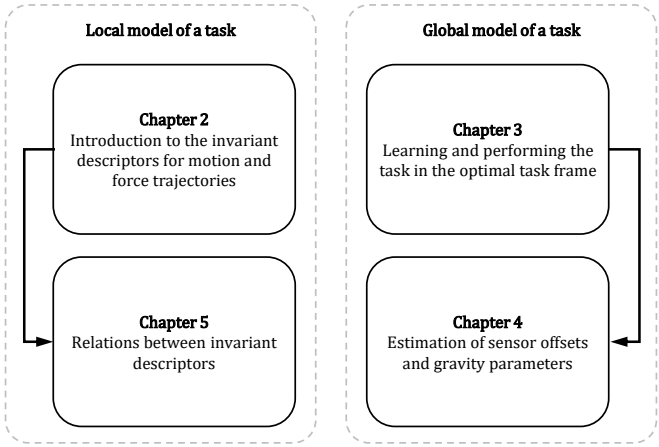


Figure 1.10: Thesis outline

Chapter 2

Invariant Descriptors of Motion and Force Trajectories for Interpreting Object Manipulation Tasks in Contact*

Abstract

Invariant descriptors of point and rigid-body motion trajectories have been proposed in the past as representative task models for motion recognition

*This chapter is based on research published as Vochten, M. and Mousavi Mohammadi, A., Verduyn, A., De Laet, T., Aertbeliën, E., and De Schutter, J. Invariant descriptors of motion and force trajectories for interpreting object manipulation tasks in contact, *IEEE Transactions on Robotics 39*, 6 (2023), 4892-4912. This publication has three primary authors, with Maxim Vochten serving as the corresponding author. He contributed to the conceptualization, investigation, software development, and drafting of the original manuscript. Ali Mousavi Mohammadi was responsible for experimental validation, data curation, investigation, software development, and the review and editing of the paper. Arno Verduyn played a key role in the methodology development, formal analysis, investigation, software development, and the review and editing of the manuscript. Tinne De Laet contributed to the early studies and investigations on force trajectories. Erwin Aertbeliën contributed to the conceptualization of the research and the review and editing of the manuscript. Joris De Schutter provided supervision, contributed to conceptualization, early studies on force trajectories, and the review and editing of the manuscript. Consequently, this work is included in this PhD thesis. The terminology and notation have been adjusted slightly for consistency throughout the thesis.

and generalization. Currently, no invariant descriptor exists for representing force trajectories, which appear in contact tasks. This chapter introduces invariant descriptors for force trajectories by exploiting the duality between motion and force. Two types of invariant descriptors are presented depending on whether the trajectories consist of screw or vector coordinates. Methods and software are provided for robustly calculating the invariant descriptors from noisy measurements using optimal control. Using experimental human demonstrations of 3D contour following and peg-on-hole alignment tasks, invariant descriptors are shown to result in task representations that do not depend on the calibration of reference frames or sensor locations. The tuning process for the optimal control problems is shown to be fast and intuitive. Similar to motions in free space, the proposed invariant descriptors for motion and force trajectories may prove useful for the recognition and generalization of constrained motions, such as during object manipulation in contact.

2.1 Introduction

In *human-robot interaction* (HRI) there is a need for generalizable task models that allow robots to reactively adapt the robot's execution and to robustly recognize human actions in continuously changing environments. For execution purposes, these models should be easily adaptable to deal with obstacles, different starting/ending/via points, robot platform constraints, and human-robot interaction. For recognition purposes, these models should be sufficiently general to recognize the human's action at different locations in space, from different viewpoints, and with different execution styles (e.g., varying scale and motion profile). The focus of this chapter is on extracting task representations that involve the manipulation of objects in contact, where not only the relative motion between objects but also interaction forces are relevant as input for the task modeling process. Typical applications include surface following tasks and assembly tasks.

Task models in HRI are commonly learned from one or multiple human demonstrations of the task using *Learning by Demonstration* [19]. The challenge is to achieve a representative task model without the need for a high number of demonstrations.

One approach to achieve representative task models from few demonstrations is to exploit invariance in the task. Here, *invariance* refers to properties of the task that remain unchanged under certain transformations such as changes in reference frame. Invariance has proven useful in various fields of robotics. Typically, the objectives were: (1) to ensure that exactly the same task execution

or behavior of a robot system was obtained within its workspace, and (2) to ensure that the quantitative characterization of such robot action using a metric resulted in exactly the same number(s) regardless of where and how the robot action was recorded or measured. Below, some examples of the importance and use of invariant solutions in different robotics contexts are briefly reported.

For the *control* of free-space robot motion such as mobile robots and UAVs, invariance was achieved by representing the trajectory tracking error in such a way that the same closed-loop behavior for trajectory tracking and disturbance rejection was obtained regardless of the direction in which the robot was moving [113]. For constrained motion, i.e., motion in contact, which is the subject of this chapter, the risk of obtaining non-invariant control solutions is more pronounced since both motion (translation and rotation) and interaction wrenches (i.e. forces and moments) have to be controlled simultaneously. For example, as pointed out in [40], it does not make sense to define *orthogonality* between two twists (i.e. translational and rotational velocities), or between two contact wrenches, because of (1) dimensional inconsistency, (2) dependence on the choice of units, and (3) dependence on the choice of the origin of the coordinates. In contrast, *reciprocity* between a twist and a wrench is an invariant concept. In [66, 67] a firm geometric foundation for kinestatics and invariant hybrid force/position control was laid based on screw theory. Taking this into consideration, invariant solutions were proposed, for example to obtain dynamic models of redundant and constrained robot manipulators [26], or in applications of hybrid force/position control [36].

In *estimation*, invariance refers to obtaining the same values for the estimated quantities regardless of the reference frame in which the estimator's states are represented. For example, the Invariant Extended Kalman Filter [16] formulated the state correction term in an invariant way using Lie group theory and showed a better convergence of the estimation compared to a standard Extended Kalman filter with a linear correction term. The Invariant Extended Kalman Filter was applied, for instance, to attitude estimation for UAVs [22].

In *motion generation*, invariance refers to obtaining exactly the same physical desired motion trajectory regardless of the reference frame in which the trajectory coordinates are expressed. Human reaching motions, represented as point trajectories, were found to correspond well to a minimum-jerk profile, an invariant property that was shown to be equivalent to having a constant equi-affine curvature [18]. Based on this property, new trajectories can be generated or adapted from reference trajectories in an invariant way [72]. This and other invariant properties such as shape preservation were considered for the deformation of trajectories using global transformations [82]. For rigid-body motion, approaches were proposed where the trajectories are either generated in a left-invariant way, meaning independent of the world reference frame, or in a

right-invariant way, meaning independent of the body reference frame [17]. This was also taken into account for the adaptation of reference or demonstrated rigid-body trajectories [61, 107].

Invariance in *action recognition* refers to recognizing actions regardless of where and how they are recorded. This can be done by comparing the recorded action instance and its model using invariant distance metrics. For example, local invariant signatures were proposed for describing and recognizing object outlines [27] and trajectories [115]. These invariant representations were shown to be invariant to occlusion, rotation, translation, and scaling.

These previous works showed that invariant representations offer many advantages that are relevant for different robotics applications. In the remainder of the chapter we focus on invariant descriptors for trajectories. These descriptors can be used to build representative invariant models of a demonstrated task in a human-robot interaction context with fewer demonstrations compared to conventional methods using coordinate-based descriptors. In addition, thanks to their invariant properties, these models enable robust action recognition in different recording contexts and exhibit good generalization properties in motion generation.

2.1.1 Invariant Trajectory Descriptors for Motion

Invariant trajectory descriptors are features of the trajectory that remain unchanged under certain transformations of the original trajectory coordinates, such as changes in reference frame, scale, and time profile. Invariant descriptors can be categorized in many types [112] depending on the features that are extracted from the curve. Many have been proposed, both for point trajectories [27, 83, 116, 90, 87] and for rigid-body trajectories [48, 47, 34, 62, 105].

This chapter focuses on a particular type of invariant descriptor, referred to as *differential invariants* [27, 112]. These invariants correspond to differential-geometric properties of the trajectory and provide a local invariant representation that is robust to occlusion and segmentation errors. Furthermore, they allow to reconstruct and generate new trajectories from the invariant descriptor by integrating corresponding differential equations. The definition of such local differential-geometric invariants can be seen as an application of *Cartan's method of moving frames* [46], which consists of defining a *natural moving frame* in the considered manifold, after which the derivative of the frame is shown to correspond to a set of geometric invariants.

For point motion in Euclidean space, the differential invariants correspond to curvature and torsion defined in a moving frame referred to as the *Frenet-Serret*

frame. Originally introduced in vision for characterizing and recognizing objects by their outlines [27, 73], the same concepts of curvature and torsion were applied for describing and recognizing point motion trajectories [115, 116].

For rigid-body motion, differential invariants were introduced based on the concept of the screw axis in screw theory [34]. A moving frame was defined on the screw axis of which the motion was described by a set of invariants referred to as *screw invariants*. These invariants are *bi-invariant*, meaning independent of both the world reference frame and the reference frame attached to the body. An alternative invariant descriptor was proposed by defining two separate Frenet-Serret frames for the position and orientation of the rigid body [62, 105], resulting in a simpler set of formulas. However, invariance for the choice of reference point for translation (the body frame's origin) was lost.

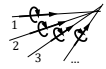
Invariant trajectory descriptors for motion have, for the large majority, been applied in motion recognition [116, 62, 105]. Trajectory generation using invariant descriptors has also been explored. The invariant descriptors are capable of both reconstructing the original trajectory [34], as well as reproducing the trajectory at new locations and in different directions by applying the correct initial values when integrating the differential equations [114]. Building further on that, trajectory adaptation methods based on optimal control and invariant descriptors were proposed [107]. The goal was to adapt a reference trajectory to comply with new constraints while minimizing deviation from the invariant descriptor of the reference trajectory. This was shown to result in excellent extrapolation capabilities [107].

Calculating differential invariants from measurement data is challenging due to noise-sensitivity and singularities. *Singularities* are defined as instances along the trajectory where components of the descriptor are arbitrarily defined. Near these singularities, small variations in the trajectory (e.g., due to measurement noise or irrelevant human variations) will result in large variations in the ill-defined components of the descriptor. To remedy this, efforts have been spent to develop robust approaches to calculate invariant descriptors, ranging from discrete approximations of the descriptor [27, 116, 62, 115] to averaging the descriptor over segments of the curve [12]. More recently, approaches based on optimal control were introduced for calculating point trajectory descriptors [78] and rigid-body trajectory descriptors [106]. Using optimal control, the local descriptor is calculated in a window of measurements while regularization terms deal with singularities and noise by smoothing the invariants in a local neighborhood. A disadvantage of current optimal control-based approaches is their dependency on many tuning parameters for weighting the trajectory accuracy cost and regularization costs, and their long calculation times.

2.1.2 Objective and Contributions

Up till now, invariant trajectory descriptors for interaction forces with similar benefits as the invariant descriptors for motion have not been proposed. This chapter's main objective is to extend the existing concepts and methodology for motion invariants towards interaction forces by exploiting duality between motion and force. The resulting invariant descriptors can be used to model motion and force trajectories that occur in motion-in-contact tasks. Table 2.1 summarizes this duality for the two cases where motion and force trajectories are either represented with vector coordinates or with screw coordinates.

Table 2.1: Overview of trajectory types of which invariant descriptors can be calculated.

Trajectory		**Applied to motion**	**Applied to force**
vectors $\boldsymbol{c}$ $\in \mathbb{R}^3$		rot. velocity $\boldsymbol{\omega}$ transl. velocity $\boldsymbol{v}$	force $\boldsymbol{f}$ moment $\boldsymbol{m}$
screws $\boldsymbol{\mathcal{S}}$ $\in \mathbb{R}^6$		*twist*: $\boldsymbol{t} = \begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{pmatrix}$	*wrench*: $\boldsymbol{w} = \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{m} \end{pmatrix}$

The main contribution of this chapter is the introduction of two types of local differential invariant descriptors for force and moment trajectories. The first type is referred to as the *vector invariant descriptor* for vector trajectories and is applied to force and moment vectors separately. The second type is referred to as the *screw axis invariant descriptor* for screw trajectories and is applied to a screw wrench, combining force and moment in one entity. The duality is shown between the new invariant descriptors for force and moment and the existing invariant descriptors for motion: the screw invariants for motion [34] and the vector invariants for translation and rotation [62, 105]. Furthermore, we explain how to calculate the invariant descriptors for motion and force using numerical optimal control schemes, which filter out the effects of measurement noise and improve the behavior of the descriptor near singularities. Triggered by the higher noise levels of measured force and moment trajectories compared to measured motion trajectories, we improved the formulation of these numerical optimal control schemes compared to [106], resulting in much more intuitive and faster tuning as well as increased robustness of the calculation. The invariant properties of the proposed descriptors were experimentally verified using a human-demonstrated 3D contour following task and a peg-on-hole alignment

Table 2.2: Overview of notation used throughout the chapter. (In this table a and b are placeholders for two frames of interest.)

Variable	Description
$\{a\}$	reference frame named *"a"* attached to a rigid body or the world
${}^{b}\boldsymbol{R}_{a}$	3×3 rotation matrix representing the orientation of frame $\{b\}$ with respect to frame $\{a\}$
${}^{b}\boldsymbol{T}_{a}$	4×4 homogeneous transformation matrix representing the orientation and position of frame $\{b\}$ with respect to frame $\{a\}$
${}_{a}\boldsymbol{\omega}$	3×1 rotational velocity vector of the rigid body expressed in $\{a\}$
${}_{a}\boldsymbol{v}^{b}$	3×1 translational velocity vector of the rigid body at the origin of $\{b\}$ and expressed in $\{a\}$
${}_{a}\boldsymbol{f}$	3×1 force vector acting on the rigid body expressed in $\{a\}$
${}_{a}\boldsymbol{m}^{b}$	3×1 moment vector acting on the rigid body at the origin of $\{b\}$ and expressed in $\{a\}$
${}_{a}\boldsymbol{\$}^{a}$	6×1 screw twist of rigid body with the first three elements the rotational velocity vector ${}_{a}\boldsymbol{\omega}$ and the last three elements the translational velocity vector ${}_{a}\boldsymbol{v}^{a}$
${}_{a}\boldsymbol{\mathcal{W}}^{a}$	6×1 screw wrench acting on the rigid body with the first three elements the force vector ${}_{a}\boldsymbol{f}$ and the last three elements the moment vector ${}_{a}\boldsymbol{m}^{a}$
${}^{b}\boldsymbol{S}_{a}$	6×6 screw transformation matrix that, when multiplied with a screw twist or wrench, transforms the screw from $\{b\}$ to $\{a\}$

task. All software used in the validation experiment, including data, has been made publicly available [7].

The remainder of the chapter is structured as follows. Section 2.2 provides the necessary background on rigid-body kinematics and statics. Sections 2.3 and 2.4 introduce local invariant descriptors in a general form, resulting in existing descriptors when applied to motion trajectories and new descriptors when applied to force and moment trajectories. Sections 2.3 and 2.4 deal with vector and screw invariant descriptors, respectively. Section 2.5 highlights a practical 3D contour following task, while Section 2.6 examines a peg-on-hole alignment task. Section 2.7 provides a discussion about the benefits and limitations of the proposed concepts and methodology, it points at future work and presents a conclusion.

2.2 Preliminaries

This section reviews essential background and introduces notation as summarized in Table 2.2.

2.2.1 Rigid-body Kinematics

The motion trajectory of a rigid body is commonly represented by attaching a reference frame $\{b\}$ to the *body* and expressing the position and orientation of this frame as a function of time t with respect to a fixed reference frame $\{w\}$, also referred to as the *world* reference frame. For all variables introduced below, the explicit dependency on time t is omitted to simplify the expressions.

The position is represented by the 3-dimensional *displacement vector* ${}^{b}\boldsymbol{p}_{w}$ from the origin of $\{w\}$ to the origin of $\{b\}$. The orientation is represented by the *rotation matrix* ${}^{b}\boldsymbol{R}_{w}$, which expresses the coordinates of the unit vectors of the moving body frame $\{b\}$ with respect to the world frame $\{w\}$:

$$ {}^{b}\boldsymbol{R}_{w} = \begin{bmatrix} \boldsymbol{e}_x & \boldsymbol{e}_y & \boldsymbol{e}_z \end{bmatrix}. \tag{2.1} $$

Position and orientation can be combined into the *homogeneous transformation matrix* or *pose matrix* ${}^{b}\boldsymbol{T}_{w}$:

$$ {}^{b}\boldsymbol{T}_{w} = \begin{bmatrix} {}^{b}\boldsymbol{R}_{w} & {}^{b}\boldsymbol{p}_{w} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix}, \tag{2.2} $$

which is part of the $SE(3)$ group according to Lie theory. The coordinates of the pose matrix ${}^{b}\boldsymbol{T}_{w}$ depend on the choice of the fixed reference frame $\{w\}$ in the world and the reference frame $\{b\}$ on the rigid body.

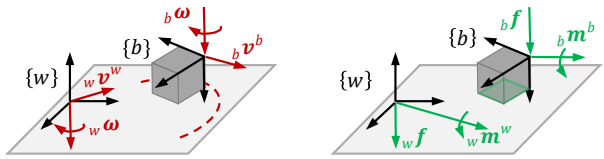
The velocity of a rigid body can be fully characterized by two 3-dimensional vectors: the rotational velocity vector $\boldsymbol{\omega}$ and the translational velocity vector $\boldsymbol{v}$ of the origin of the considered reference frame. These vectors are summarized with a 6-dimensional screw [15], the *twist* $\boldsymbol{t} = \begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{pmatrix}$.

To define the time-derivative of the homogeneous transformation matrix, $\dot{\boldsymbol{T}} = \frac{\mathrm{d}\boldsymbol{T}}{\mathrm{d}t}$, we first introduce the operator $[\cdot]_{\times}$. When applied to the vector $\boldsymbol{\omega} = [\omega_x \ \omega_y \ \omega_z]^T$, it results in a 3×3 skew-symmetric matrix:

$$ [\boldsymbol{\omega}]_{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}, \tag{2.3} $$

and when applied to a screw $\boldsymbol{t}$, it results in the following 4×4 *twist matrix* $[\boldsymbol{t}]_{\times}$, which is an element of the Lie algebra $se(3)$:

$$ [\boldsymbol{t}]_{\times} = \begin{bmatrix} [\boldsymbol{\omega}]_{\times} & \boldsymbol{v} \\ \boldsymbol{0}_{1\times 3} & 0 \end{bmatrix}. \tag{2.4} $$



(a) Screw twists (b) Screw wrenches

Figure 2.1: Visualization of screw twist $\boldsymbol{t} = (\boldsymbol{\omega}^T \ \boldsymbol{v}^T)^T$ and screw wrench $\boldsymbol{w} = (\boldsymbol{f}^T \ \boldsymbol{m}^T)^T$ in the world frame $\{w\}$ and in the body frame $\{b\}$.

The definition of the time-derivative of the homogeneous transformation matrix $\dot{\boldsymbol{T}}$ depends on the choice of reference frame in which the twist $\boldsymbol{t}$ is expressed. Choosing the body frame $\{b\}$, the *body twist* ${}_b\boldsymbol{t}^b$ contains the rotational velocity vector ${}_b\boldsymbol{\omega}$ expressed in $\{b\}$ and the translational velocity vector ${}_b\boldsymbol{v}^b$ of the origin of $\{b\}$ and expressed in $\{b\}$ (see Figure 2.1a). The derivative of the pose matrix ${}^b\dot{\boldsymbol{T}}_w$ is then defined as:

$$ {}^b\dot{\boldsymbol{T}}_w = {}^b\boldsymbol{T}_w \left[{}_b\boldsymbol{t}^b \right]_\times, \text{ with } \left[{}_b\boldsymbol{t}^b \right]_\times = \begin{bmatrix} [{}_b\boldsymbol{\omega}]_\times & {}_b\boldsymbol{v}^b \\ \boldsymbol{0}_{1\times 3} & 0 \end{bmatrix}. \tag{2.5} $$

The twist ${}_b\boldsymbol{t}^b$ has the property of being invariant for a change of $\{w\}$ (*left-invariance*).

Alternatively, choosing the world frame $\{w\}$, the spatial twist ${}_w\boldsymbol{t}^w$ consists of the rotational velocity vector ${}_w\boldsymbol{\omega}$ expressed in $\{w\}$, and the translational velocity vector ${}_w\boldsymbol{v}^w$ of the point on the rigid body that instantaneously coincides with the origin of $\{w\}$ expressed in $\{w\}$. The pose derivative ${}^b\dot{\boldsymbol{T}}_w$ is then defined as:

$$ {}^b\dot{\boldsymbol{T}}_w = \left[{}_w\boldsymbol{t}^w \right]_\times {}^b\boldsymbol{T}_w, \text{ with } \left[{}_w\boldsymbol{t}^w \right]_\times = \begin{bmatrix} [{}_w\boldsymbol{\omega}]_\times & {}_w\boldsymbol{v}^w \\ \boldsymbol{0}_{1\times 3} & 0 \end{bmatrix}. \tag{2.6} $$

where $\left[{}_w\boldsymbol{t}^w \right]_\times$ is now multiplied on the left of ${}^b\boldsymbol{T}_w$ since the twist is expressed in $\{w\}$. The twist ${}_w\boldsymbol{t}^w$ has the property of being invariant for a change of $\{b\}$ (*right-invariance*).

The relation between twists ${}_w\boldsymbol{t}^w$ and ${}_b\boldsymbol{t}^b$ is provided by the 6×6 *screw transformation matrix* $\boldsymbol{S}({}^b\boldsymbol{T}_w)$ (also referred to as the *adjoint matrix* [76]), which is constructed from elements of the pose matrix ${}^b\boldsymbol{T}_w$:

$$ {}_w\boldsymbol{t}^w = \boldsymbol{S}({}^b\boldsymbol{T}_w) \ {}_b\boldsymbol{t}^b = \begin{bmatrix} {}^b\boldsymbol{R}_w & \boldsymbol{0}_{3\times 3} \\ [{}^b\boldsymbol{p}_w]_\times {}^b\boldsymbol{R}_w & {}^b\boldsymbol{R}_w \end{bmatrix} {}_b\boldsymbol{t}^b. \tag{2.7} $$

Twists ${}_b\boldsymbol{t}^b$ and ${}_w\boldsymbol{t}^w$ represent first-order kinematics of the trajectory and are invariant to the choice of world frame and body frame, respectively. But neither

${}_w\boldsymbol{t}^w$ nor ${}_b\boldsymbol{t}^b$ is invariant for the choice of *both* reference frames $\{w\}$ and $\{b\}$. Trajectory descriptors that are invariant to *both* reference frames (also referred to as coordinate-invariance) will be discussed in Sections 2.3 and 2.4.

2.2.2 Duality between Twist and Wrench as General Screws

All forces and moments acting on a rigid body can be reduced to a single resultant force vector and a single resultant moment vector. One option is to express these vectors in the body frame: ${}_b\boldsymbol{f}$ represents the resultant force expressed in $\{b\}$, and ${}_b\boldsymbol{m}^b$ represents the moment with respect to the origin of $\{b\}$ and with coordinates expressed in $\{b\}$ (see Figure 2.1b). Similar to the *twist*, a compact notation for the resulting force and moment is the 6-dimensional *screw wrench*:

$$ {}_b\boldsymbol{w}^b = \begin{pmatrix} {}_b\boldsymbol{f} \\ {}_b\boldsymbol{m}^b \end{pmatrix}. \tag{2.8} $$

Transforming the wrench from the body frame $\{b\}$ to the world frame $\{w\}$ is done using the same 6×6 *screw transformation matrix* $\boldsymbol{S}({}^b\boldsymbol{T}_w)$ as for the twist:

$$ {}_w\boldsymbol{w}^w = \boldsymbol{S}({}^b\boldsymbol{T}_w)\ {}_b\boldsymbol{w}^b. \tag{2.9} $$

This analogy between twists and wrenches is also referred to as the *duality between velocity and force*. Both twist and wrench can be interpreted as a 6-dimensional *screw* [15], meaning they can always be represented by a vector lying along a line in space and the moment of that vector around the line.

2.2.3 Time-invariance Using a Geometric Progress Variable

In the previous subsections, motion and force trajectories were assumed to be a function of time t. To obtain a *time-invariant trajectory representation*, i.e., a trajectory representation that is invariant with respect to the motion profile, a geometric progress variable ξ must be chosen.

For tasks where translation is dominant, such as in contour following, the arc length is a natural choice. The arc length is calculated by integrating the norm of the translational velocity vector $\boldsymbol{v}$ over time. For tasks where rotation is dominant, such as in opening a door, a natural progress variable would be the integral of the norm of the rotational velocity vector $\boldsymbol{\omega}$. If translation and rotation are both important, then a progress parameter can be defined combining the translation and rotation, for example, using a linear combination [34]:

$$ \xi(t) = \int_0^t w \left\|\boldsymbol{\omega}\right\| + (1-w) \left\|\boldsymbol{v}\right\| \mathrm{d}t, \tag{2.10} $$

where w weights the relative contribution. In [34], further details are provided for obtaining a progress value that is invariant for the choice of reference point on the body for the translation $\boldsymbol{v}$.

For the pose $\boldsymbol{T}$ and wrench $\boldsymbol{w}$, the reparameterization from time t to progress ξ is straightforward:

$$\boldsymbol{T}(\xi) = \boldsymbol{T}(t(\xi)) \quad \text{and} \quad \boldsymbol{w}(\xi) = \boldsymbol{w}(t(\xi)). \tag{2.11}$$

In practice, since measurement data is often only available in a discretely sampled form, the reparameterization is achieved through numerical interpolation.

For the velocity, the reparameterization from time t to progress ξ requires an additional step since velocities represent derivatives of the trajectory. As an example, the reparameterization of the translational velocity vector $\boldsymbol{v}$, as the derivative of the position $\boldsymbol{p}$, can be worked out as:

$$\boldsymbol{v}(\xi) = \frac{\mathrm{d}\boldsymbol{p}(\xi)}{\mathrm{d}\xi} = \frac{\mathrm{d}\boldsymbol{p}(t(\xi))}{\mathrm{d}t(\xi)} \cdot \frac{\mathrm{d}t(\xi)}{\mathrm{d}\xi} = \frac{\boldsymbol{v}(t(\xi))}{\dot{\xi}(t(\xi))}, \tag{2.12}$$

where the chain rule of differentiation was applied in the second equality to obtain the relation with time t, and the *progress rate* $\dot{\xi} = \frac{\mathrm{d}\xi}{\mathrm{d}t}$ is the derivative of the progress with respect to time. Similar derivations can be made for the rotational velocity vector $\boldsymbol{\omega}$ and the twist $\boldsymbol{t}$:

$$\boldsymbol{\omega}(\xi) = \frac{\boldsymbol{\omega}(t(\xi))}{\dot{\xi}(t(\xi))} \quad \text{and} \quad \boldsymbol{t}(\xi) = \frac{\boldsymbol{t}(t(\xi))}{\dot{\xi}(t(\xi))}. \tag{2.13}$$

Equations (2.11)-(2.13) can be used in the other direction to apply a motion profile $\xi(t)$ and obtain the pose, wrench, translational velocity, rotational velocity or twist as a function of time. This is relevant, for example, when a trajectory was planned in the progress domain ξ and needs to be executed with a specific timing $\xi(t)$. A special choice for the progress variable is one where $\xi = 0$ at the beginning of the trajectory and $\xi = 1$ at the end of the trajectory. This results in a reparameterization that is independent of the scale of the trajectory [34].

2.3 Invariant Descriptor for Vector Trajectories

This section derives a local invariant descriptor for a general *vector trajectory* $\boldsymbol{c}(\xi)$. The trajectory $\boldsymbol{c}(\xi)$ is assumed to be time-invariant, i.e. parameterized as a function of a geometric progress ξ as discussed in Section 2.2.3. To keep the formulas concise, the explicit dependency on ξ is omitted in the following formulas. The derivative of $\boldsymbol{c}$ with respect to the progress ξ is represented using the prime notation: $\boldsymbol{c}' = \frac{\mathrm{d}\boldsymbol{c}}{\mathrm{d}\xi}$.

2.3.1 Definition of Invariants

The vector invariants are based on a generalization of the *Frenet-Serret equations* for translation [45], where $\boldsymbol{c}$ corresponded to the translational velocity vector. While in [105, 62] the Frenet-Serret equations were extended to rotational velocity trajectories, the aim here is to exploit the duality between velocity and force so that the invariants can be presented in a general form, applicable to rotational and translational velocity trajectories, as well as to force and moment trajectories.

The vector invariants are defined and expressed in a local moving frame, referred to as the *Frenet-Serret frame* (FS frame), which is constructed as follows. Its first axis, the tangent $\boldsymbol{e}_t$, corresponds to the unit vector in the direction of $\boldsymbol{c}$. The second and third axes correspond to the unit vectors $\boldsymbol{e}_n$ and $\boldsymbol{e}_b$ which are referred to as the normal and binormal axes of the trajectory, respectively. These unit vectors are defined as follows from $\boldsymbol{c}$ and $\boldsymbol{c}'$[2]:

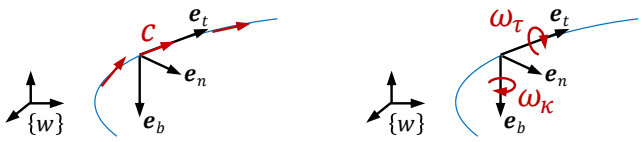
$$\boldsymbol{e}_t = \frac{\boldsymbol{c}}{\|\boldsymbol{c}\|}, \quad \boldsymbol{e}_n = \frac{(\boldsymbol{c} \times \boldsymbol{c}') \times \boldsymbol{c}}{\|(\boldsymbol{c} \times \boldsymbol{c}') \times \boldsymbol{c}\|}, \quad \boldsymbol{e}_b = \boldsymbol{e}_t \times \boldsymbol{e}_n. \tag{2.14}$$

In differential geometry, the linear span of $\boldsymbol{e}_t$ and $\boldsymbol{e}_n$ is referred to as the *osculating plane*, i.e. the plane in which the vector $\boldsymbol{c}$ rotates instantaneously, while $\boldsymbol{e}_b$ represents the axis about which this rotation occurs. The complete orientation of the FS frame is expressed using a rotation matrix $\widetilde{\boldsymbol{R}}$:

$$\widetilde{\boldsymbol{R}} = \begin{bmatrix} \boldsymbol{e}_t & \boldsymbol{e}_n & \boldsymbol{e}_b \end{bmatrix}, \tag{2.15}$$

where the tilde $\sim$ indicates it is a local moving frame, in this case the FS frame. See Figure 2.2 for a visualization of the frame.

[2]To be consistent with the common notation of the FS frame and equations, the authors adapted the order of the columns $\boldsymbol{e}_t$, $\boldsymbol{e}_n$, and $\boldsymbol{e}_b$ of $\widetilde{\boldsymbol{R}}$ in their previous papers such as [34, 106, 107]. Therefore, this chapter has a slightly different notation than the authors' previous works. This adaptation affects only the appearance of the equations, not the concept. For visualization of this adaption and comparing the notations, please refer to Appendix A.



(a) Object invariant along tangent (b) Moving frame invariants

Figure 2.2: Invariant descriptor for a vector trajectory: (a) object invariant c defined in the moving FS frame given by the unit vectors $\boldsymbol{e}_t$, $\boldsymbol{e}_n$ and $\boldsymbol{e}_b$; (b) moving frame invariants ω_τ and ω_κ. The figure depicts the special case where $\boldsymbol{c}$ corresponds to a translational velocity vector with respect to reference frame $\{w\}$ and where, for visualization purposes, the moving frame is depicted with its origin chosen at the corresponding position along the point trajectory (indicated in blue).

The invariants are then defined as follows. The first invariant c is the magnitude of the vector $\boldsymbol{c}$ along the tangent:

$$c = \boldsymbol{c} \cdot \boldsymbol{e}_t. \tag{2.16}$$

This invariant is referred to as the *object invariant* since it relates to the object. For example, c can represent the magnitude of the translational or rotational velocity vector of the object, or the magnitude of the force or moment vector applied to the object. The vector $\boldsymbol{c}$ can be retrieved from the object invariant c using:

$$\boldsymbol{c} = \widetilde{\boldsymbol{R}}\, \widetilde{\boldsymbol{c}}, \quad \text{with} \quad \widetilde{\boldsymbol{c}} = \begin{pmatrix} c & 0 & 0 \end{pmatrix}^T. \tag{2.17}$$

The second and third invariants ω_κ and ω_τ govern the first-order kinematics of the local FS frame[3]:

$$\widetilde{\boldsymbol{R}}' = \widetilde{\boldsymbol{R}} \begin{bmatrix} 0 & -\omega_\kappa & 0 \\ \omega_\kappa & 0 & -\omega_\tau \\ 0 & \omega_\tau & 0 \end{bmatrix}, \tag{2.18}$$

where ω_κ is known as the curvature rate and ω_τ as the torsion rate, with the following explicit formulas [45]:

$$\omega_\kappa = \frac{||\boldsymbol{c} \times \boldsymbol{c}'||}{||\boldsymbol{c}||^2}, \qquad \omega_\tau = \frac{((\boldsymbol{c} \times \boldsymbol{c}') \times (\boldsymbol{c} \times \boldsymbol{c}'')) \cdot \boldsymbol{c}}{||\boldsymbol{c} \times \boldsymbol{c}'||^2 ||\boldsymbol{c}||}. \tag{2.19}$$

For a naturally parameterized point trajectory, with $\boldsymbol{c}$ the point's velocity, ω_κ and ω_τ correspond to the local curvature κ and torsion τ of the point's trajectory.

[3]The relation between (2.18) and the Frenet-Serret equation as usually presented in literature is derived in Appendix B.

Equation (2.18) can be written compactly using (2.3):

$$\widetilde{\boldsymbol{R}}' = \widetilde{\boldsymbol{R}} \left[\boldsymbol{i}\right]_\times, \quad \text{with} \quad \boldsymbol{i} = \begin{pmatrix} \omega_\tau \\ 0 \\ \omega_\kappa \end{pmatrix}, \tag{2.20}$$

where $\boldsymbol{i}$ is referred to as the *moving frame invariant* since it models the first-order kinematics of the moving frame.

Only the orientation is relevant for the FS frame, not its position. Therefore it is sometimes referred to as an *orientation frame* [33]. For visualization purposes, its evolution is often shown by representing it at the point on the motion trajectory associated with the progress variable ξ. This was done in Figure 2.2, which shows the definition of the moving frame and the corresponding invariants for a point trajectory.

Singularities: Singularities are defined as instances where axes of the moving frame and the corresponding invariants are not uniquely defined anymore. A first type of singularity is when $\boldsymbol{c} = \boldsymbol{0}$. The object invariant c is then zero, while the moving frame invariants ω_κ and ω_τ are undefined. In other words, the complete moving frame is arbitrary.

A second type of singularity is when the vector $\boldsymbol{c}$ has an unchanging orientation at some instance ($\boldsymbol{c} \times \boldsymbol{c}' = \boldsymbol{0}$), remaining parallel to itself. The first moving frame invariant ω_κ is then zero while the second moving frame invariant ω_τ is undefined. In other words, the tangent of the moving frame is well-defined, but the normal and binormal axes are arbitrary.

Invariant properties: If the trajectory $\boldsymbol{c}(\xi)$ is expressed as a function of a geometric progress variable ξ as explained in Section 2.2.3, then the descriptor $(c, \boldsymbol{i})$ is also *time-invariant*. Furthermore, the descriptor is *invariant with respect to the choice of world frame* $\{w\}$ (both origin and orientation) in which the coordinates of the given vector trajectory $\boldsymbol{c}(\xi)$ are expressed. It is evident that a change of $\{w\}$ merely results in the same change of orientation for the vectors $\boldsymbol{c}$, $\boldsymbol{c}'$, and $\boldsymbol{c}''$, resulting in the same object invariant c, the same FS frame for all ξ, and hence also the same moving frame invariant $\boldsymbol{i}$. If the coordinates of $\boldsymbol{c}$ are instead expressed in $\{b\}$, the descriptor is still *invariant for changes in orientation of the body frame* $\{b\}$. However, not all considered instances of the descriptor are invariant with respect to the origin of body frame $\{b\}$. As pointed out in [105], the translational velocity vector $\boldsymbol{v}$ depends on the reference point on the moving object chosen to express its velocity, hence on the origin of $\{b\}$. Similarly, the moment vector $\boldsymbol{m}$ depends on the reference point for expressing the moment acting on the object, hence on the origin of $\{b\}$. The dependency of $\boldsymbol{v}$ and $\boldsymbol{m}$, and hence also of their invariant descriptors, on the origin of $\{b\}$ is a disadvantage compared to the screw invariants discussed in Section 2.4, which

do not exhibit this dependency on the chosen reference point on the moving body.

Trajectory reconstruction: Reconstructing the trajectory $\boldsymbol{c}(\xi)$ from the invariant descriptor $(\widetilde{\boldsymbol{c}}(\xi), \boldsymbol{i}(\xi))$ requires integrating the differential equation (2.20) from an initial value of the moving frame $\widetilde{\boldsymbol{R}}$. In general, there exists no closed-form solution for the integration. Instead, the problem is discretized so that $\widetilde{\boldsymbol{R}}_{k+1}$ at sample $k+1$ is found from $\widetilde{\boldsymbol{R}}_k$ at sample k as follows:

$$\widetilde{\boldsymbol{R}}_{k+1} = \widetilde{\boldsymbol{R}}_k \ \exp\left([\boldsymbol{i}_k]_\times \Delta\xi\right), \tag{2.21}$$

where we assume that $\boldsymbol{i}_k$ remains constant over the integration step $\Delta\xi$. The matrix exponential operator $\exp(\cdot)$ maps the skew-symmetric matrix $[\boldsymbol{i}_k]_\times \Delta\xi$, which is part of the Lie algebra $so(3)$, into the corresponding change in orientation, which is part of the Lie group $SO(3)$. An explicit expression of the matrix exponential can be formulated using Rodrigues' rotation formula [76]. Given the initial value $\widetilde{\boldsymbol{R}}_1$ at $k=1$, the moving frame can be reconstructed for all remaining samples $k = 2...N$. From the moving frames $\widetilde{\boldsymbol{R}}_k$, the corresponding vector trajectory $\boldsymbol{c}_k$ at $k : 1...N$ is found using (2.17):

$$\boldsymbol{c}_k = \widetilde{\boldsymbol{R}}_k \ \widetilde{\boldsymbol{c}}_k, \quad \forall k \in [1, N]. \tag{2.22}$$

Equations (2.21) and (2.22) are referred to as the *moving frame equation* and *object trajectory equation*, respectively. Table 2.3 on page 43 provides an overview of the vector invariants framework with application to translational velocity $\boldsymbol{v}$, rotational velocity $\boldsymbol{\omega}$, force $\boldsymbol{f}$, and moment $\boldsymbol{m}$ trajectories.

2.3.2 Robust Calculation of Invariants Using Optimal Control

Using the analytical formulas to calculate the vector invariants c and $\boldsymbol{i}$ has three pitfalls: (1) near singularities, some of the invariants and axes of the moving frame $\widetilde{\boldsymbol{R}}$ are ill-defined, hence the behavior of c and $\boldsymbol{i}$ is not well-captured in such case; (2) the high-order derivatives cause sensitivity to noise, especially near singularities; and (3) the open-loop reconstructed trajectory using (2.21)-(2.22) is not guaranteed to match the original trajectory due to integration drift.

In addition, the invariants c and ω_κ are always positive by definition from (2.16)-(2.19) resulting in *frame flipping* issues: the FS frame must flip 180° when the direction of $\boldsymbol{c}$ reverses or when the curve traced by the vector $\boldsymbol{c}$ has an inflection point. Other definitions of moving frames exist to minimize the frame's rotation, such as Bishop frames [21], but these frames are not locally defined anymore.

To deal with these problems, we formulate an *Optimal Control Problem* (OCP) to calculate the vector invariants c and $\boldsymbol{i}$ in a robust way. An OCP is a type of optimization problem that, for a given dynamical system, tries to find control inputs that minimize a specific cost function over a horizon. Here, the objective of the optimization problem is to find vector invariants that reconstruct the given measured vector trajectory. The latter can be quantified by introducing a trajectory reconstruction cost $\Delta \boldsymbol{c}_{\text{MSE}}$ containing the mean-squared error (MSE) between the reconstructed trajectory $\boldsymbol{c}_k$ and measured trajectory $\boldsymbol{c}_k^{meas}$ for $k = 1...N$:

$$\Delta \boldsymbol{c}_{\text{MSE}}(\boldsymbol{c}, \boldsymbol{c}^{meas}) = \frac{1}{N} \sum_{k=1}^{N} \|\boldsymbol{c}_k - \boldsymbol{c}_k^{meas}\|^2. \tag{2.23}$$

To deal with measurement noise and the effects of singularities, a regularization cost is introduced. In general, the regularization cost can be any function of invariants, including derivatives of invariants. In [106], the terms in the regularization cost were chosen as the derivatives of all invariants and the absolute values of the moving frame invariants. The trajectory reconstruction cost and regularization cost were added in the objective function of the OCP, and as a result they had to be weighted with appropriate weighting factors, requiring an extensive tuning process.

To reduce tuning efforts and improve interpretability of chosen parameters, this chapter proposes a new formulation of the OCP. The trajectory reconstruction cost is moved to the constraints of the OCP and is bounded by a chosen tolerance value ϵ. For vector trajectories, this is worked out as follows:

$$\epsilon_c^2 \geq \Delta \boldsymbol{c}_{\text{MSE}}(\boldsymbol{c}, \boldsymbol{c}^{meas}), \tag{2.24}$$

in which ϵ_c is interpreted as the desired tolerance on the trajectory reconstruction cost over the entire horizon. The subscript in ϵ_c refers to the type of trajectory that is considered. The value of ϵ can be set by the user based on the expected accuracy of the measurement system or the tolerable reconstruction error for the given application.

For the regularization term in the objective function, we take the squared norm of the moving frame invariants $\boldsymbol{i}$. Since $\boldsymbol{i}$ signifies the rotational change of the moving frame, this will result in a more stable and smooth evolution of the moving frame $\widetilde{\boldsymbol{R}}$. To avoid frame flipping issues, all invariants are allowed to become positive or negative, resulting in a smoother evolution of the FS frame.

The complete OCP to calculate the vector invariants $\widetilde{\boldsymbol{c}}$ and $\boldsymbol{i}$ from a given sequence of N measured vectors $\boldsymbol{c}_k^{meas}$ is:

$$\underset{\widetilde{c}_{[\cdot]},\boldsymbol{i}_{[\cdot]},\boldsymbol{c}_{[\cdot]},\widetilde{\boldsymbol{R}}_{[\cdot]}}{\text{minimize}} \quad \sum_{k=1}^{N-1} \|\boldsymbol{i}_k\|^2, \tag{2.25}$$

subject to:

$$\epsilon_c^2 \geq \Delta \boldsymbol{c}_{\text{MSE}}(\boldsymbol{c}, \boldsymbol{c}^{meas}), \tag{r.2.24}$$

$$\widetilde{\boldsymbol{R}}_{k+1} = \widetilde{\boldsymbol{R}}_k \ \exp\left([\boldsymbol{i}_k]_\times \Delta\xi\right), \quad \forall k \in [1, N-1] \tag{r.2.21}$$

$$\boldsymbol{c}_k = \widetilde{\boldsymbol{R}}_k \ \widetilde{\boldsymbol{c}}_k, \qquad \forall k \in [1, N] \tag{r.2.22}$$

$$\widetilde{\boldsymbol{R}}_1^T \widetilde{\boldsymbol{R}}_1 = \boldsymbol{I}_{3\times 3}. \tag{2.26}$$

Besides the trajectory error constraint (2.24), the constraints consist of the trajectory reconstruction equations (2.21)-(2.22), while constraint (2.26) imposes orthonormality conditions on the initial moving frame.[4] The orthonormality of $\widetilde{\boldsymbol{R}}$ is preserved throughout the entire horizon through the matrix exponential integrator in (2.21).

Initialization: The OCP (2.25)-(2.26) is a non-convex optimization problem and therefore multiple local minima may exist. The solution will converge towards one of the local minima depending on the initialization of the unknown variables. To obtain invariant solutions regardless of the reference frame in which the variables are represented, the variables must be initialized in an invariant way. One possibility is to calculate initial values for the invariants and moving frames using the analytical formulas (2.15)-(2.19), as was done in [106]. However, this initialization then suffers from the same three pitfalls mentioned at the beginning of this subsection.

We propose an alternative initialization approach here based on calculating an average moving frame, first introduced in [10] for screw trajectories, but

[4]Enforcing this constraint in practice remains a difficult and open problem. While the goal is to ensure orthonormality, the constraint introduces nine scalar conditions, some of which are redundant. One solution is to apply the constraint only to the upper or lower triangular part of the rotation matrix, thereby reducing redundancy. However, in certain cases, using the full set of constraints results in better convergence. Another option is to represent rotations using quaternions. Despite these options, no single method consistently outperforms the others across all scenarios. As a result, we mainly rely on either the full or triangular constraint formulations. It is also important to note that explicitly enforcing orthonormality on the initial moving frame is sufficient, as it is assumed to be initialized with reliable values that already satisfy this condition.

adapted here to vector trajectories in three steps. First, calculate the covariance matrix $\boldsymbol{C}_c$ associated with the measured vector trajectory $\boldsymbol{c}_k^{meas}$:

$$\boldsymbol{C}_c = \frac{1}{N} \sum_{k=1}^{N} \boldsymbol{c}_k^{meas} \left(\boldsymbol{c}_k^{meas}\right)^T . \tag{2.27}$$

Second, the orthogonal matrix of eigenvectors $\boldsymbol{U}_c$ is calculated from $\boldsymbol{C}_c$. This matrix $\boldsymbol{U}_c$ is invariant for changes in reference frame by definition and can be interpreted as a sort of average FS frame for the whole trajectory. The signs of the first two eigenvectors are chosen unambiguously by making the average value of the measurement vector $\boldsymbol{c}_k^{meas}$ positive along these directions. The third axis then follows from the cross product of the first two axes. All moving frames $\widetilde{\boldsymbol{R}}_k$ are initialized with the average moving frame $\boldsymbol{U}_c$.

Third, to be consistent with the average moving frame, the moving frame invariants ω_κ and ω_τ are initialized with zero values. The object invariant c_k is initialized by projecting the vector trajectory $\boldsymbol{c}_k^{meas}$ onto the average moving frame and selecting the component along the first axis for all samples k.

Application to motion and force data: The calculation of vector invariants using OCP (2.25)-(2.26) can directly be applied to translational velocity trajectories $\boldsymbol{v}(\xi)$, rotational velocity trajectories $\boldsymbol{\omega}(\xi)$, force trajectories $\boldsymbol{f}(\xi)$, and moment trajectories $\boldsymbol{m}(\xi)$.

However, in many applications involving rigid-body motion, the measured data corresponds to position and orientation coordinates of the rigid body, instead of translational and rotational velocity vectors. In this case, one option is to use the following two-step approach: (1) estimate the rigid-body velocities from the measured position and orientation coordinates; (2) apply OCP (2.25)-(2.26) to the estimated velocity vectors. A downside of this first option is that integration of the optimized velocity vectors to reconstruct position and orientation coordinates will inevitably result in integration drift, such that there will be a deviation between the measured and reconstructed positions and orientations.

A second option is to adapt optimization problem (2.25)-(2.26) to include position or orientation measurements. Below, this is worked out assuming the measured positions are given by position vectors and the measured orientations by rotation matrices. The proposed approach can be adapted to deal with other orientation representations such as quaternions.

Including position measurements for case $\boldsymbol{c} = \boldsymbol{v}$: The reconstructed position vectors $\boldsymbol{p}_k$ at each sample are introduced as additional variables in the OCP. The trajectory reconstruction cost $\Delta \boldsymbol{c}_{\text{MSE}}(\boldsymbol{c}, \boldsymbol{c}^{meas})$ in (2.23) is now replaced by the MSE error between the reconstructed position $\boldsymbol{p}_k$ and the

measured position $\boldsymbol{p}_k^{meas}$ over $k = 1...N$:

$$\Delta \boldsymbol{p}_{\mathrm{MSE}}(\boldsymbol{p}, \boldsymbol{p}^{meas}) = \frac{1}{N} \sum_{k=1}^{N} \|\boldsymbol{p}_k - \boldsymbol{p}_k^{meas}\|^2. \tag{2.28}$$

The reconstructed positions are defined by extending the integrator in (2.21) with a fourth row and column:

$$\begin{bmatrix} \widetilde{\boldsymbol{R}}_{k+1} & \boldsymbol{p}_{k+1} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} = \begin{bmatrix} \widetilde{\boldsymbol{R}}_{k} & \boldsymbol{p}_{k} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \exp\left(\begin{bmatrix} [\boldsymbol{i}_k]_\times & \widetilde{\boldsymbol{c}}_k \\ \boldsymbol{0}_{1\times 3} & 0 \end{bmatrix} \Delta\xi \right). \tag{2.29}$$

The effect of these additions is that $\widetilde{\boldsymbol{c}}_k$, which in this case signifies the translational velocity in the moving frame, is used to update the position $\boldsymbol{p}_k$.

Including orientation measurements for case $\boldsymbol{c} = \boldsymbol{\omega}$: The workflow is similar as for position. Variables for the reconstructed rotation matrices $\boldsymbol{R}_k$ at each sample are introduced. The trajectory reconstruction cost $\Delta \boldsymbol{c}_{\mathrm{MSE}}(\boldsymbol{c}, \boldsymbol{c}^{meas})$ in (2.23) is replaced by a first-order approximation of the mean-squared rotation angle [52] between the reconstructed rotation matrix $\boldsymbol{R}_k$ and the measured rotation matrix $\boldsymbol{R}_k^{meas}$ over $k = 1...N$:

$$\Delta \boldsymbol{R}_{\mathrm{MSE}}(\boldsymbol{R}, \boldsymbol{R}^{meas}) = \frac{1}{N} \sum_{k=1}^{N} \left\| (\boldsymbol{R}_k^{meas})^T \boldsymbol{R}_k - \boldsymbol{I}_{3\times 3} \right\|_{F_u}^2, \tag{2.30}$$

where $\|\cdot\|_{F_u}$ corresponds to a Frobenius norm that is applied to the upper-triangular part of the matrix. The reconstructed rotation matrices $\boldsymbol{R}_k$ are defined by integrating $\boldsymbol{c}_k$, which now signifies the rotational velocity, in an additional reconstruction constraint:

$$\boldsymbol{R}_{k+1} = \exp\left([\boldsymbol{c}_k]_\times \Delta\xi\right) \boldsymbol{R}_k, \qquad \forall k \in [1, N]. \tag{2.31}$$

Finally, adding the constraint $\boldsymbol{R}_1^T \boldsymbol{R}_1 = \boldsymbol{I}_{3\times 3}$ ensures that orthonormality is preserved throughout the trajectory.[5]

2.4 Invariant Descriptor for Screw Trajectories

This section derives a local invariant descriptor for a screw trajectory $\boldsymbol{\jmath}(\xi)$, parameterized with respect to a progress variable ξ:

$$\boldsymbol{\jmath}(\xi) = \begin{pmatrix} \boldsymbol{a}(\xi) \\ \boldsymbol{b}(\xi) \end{pmatrix}, \quad \text{with} \quad \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^{3\times 1}, \tag{2.32}$$

[5] The same considerations discussed for (2.26) also apply in this case.

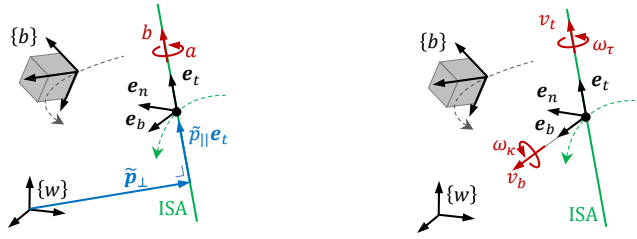
where $\boldsymbol{a}$ is the direction part of the screw and $\boldsymbol{b}$ the moment part. Again we omit the explicit dependency on ξ for conciseness while the derivative of $\boldsymbol{\jmath}$ with respect to the progress ξ is denoted as: $\boldsymbol{\jmath}' = \frac{\mathrm{d}\boldsymbol{\jmath}}{\mathrm{d}\xi}$.

2.4.1 Definition of Invariants

The screw invariants are based on a generalization of *screw theory* for rigid-body kinematics and statics. In 1900, Ball [15] published a complete *theory of screws* based on the theorems of Chasles and Poinsot. *Chasles' theorem* states that any rigid-body displacement can always be represented as a rotation and translation along an axis in space. *Poinsot's theorem* is the dual to Chasles' theorem and it states that all forces and moments acting upon a rigid body can always be replaced by a single force and moment along an axis in space. This duality allows us to calculate the screw invariants of a wrench trajectory similar to how it was done for a twist trajectory in [34]. Hence, $\boldsymbol{\jmath}$ can represent either a twist $\boldsymbol{t}$ or a wrench $\boldsymbol{w}$. In screw theory, the *screw axis* refers to the axis along which the twist or wrench can be expressed [42], but commonly it is also referred to as the *instantaneous screw axis* (ISA) to highlight its instantaneous nature.

The screw invariants are expressed in a local moving frame, referred to as the *Instantaneous Screw Axis frame* (ISA frame), which is constructed as follows. The orientation of the ISA frame is completely determined by $\boldsymbol{a}$, the direction part of the screw, i.e. $\boldsymbol{\omega}$ for a twist and $\boldsymbol{f}$ for a wrench. Hence, this orientation, represented by the rotation matrix $\widetilde{\boldsymbol{R}}$, is calculated in a similar way as the FS frame using (2.15) after replacing $\boldsymbol{c}$ by $\boldsymbol{a}$. Now, the tangent $\boldsymbol{e}_t$ is along the ISA, while the binormal $\boldsymbol{e}_b$ is in the direction in which the ISA is instantaneously rotating.

On the other hand, the origin of the ISA frame $\widetilde{\boldsymbol{p}}$ is defined as the point on the ISA about which the ISA is instantaneously rotating. An analytic procedure to derive this origin was proposed in [34]. We summarize it here as a two-step procedure, illustrate it schematically in Figure 2.3a, and refer to [34] for more detail.



(a) Object invariants (b) Moving frame invariants

Figure 2.3: Invariant descriptor for a screw trajectory: (a) object invariants a and b defined in the moving ISA frame given by the unit vectors $\boldsymbol{e}_t$, $\boldsymbol{e}_n$, $\boldsymbol{e}_b$, and the origin $\widetilde{\boldsymbol{p}} = \widetilde{\boldsymbol{p}}_\perp + \widetilde{p}_\parallel \boldsymbol{e}_t$; (b) the moving frame invariants ω_κ, ω_τ, v_b, and v_t. The figure shows the special case where the screw $\boldsymbol{\$}$ corresponds to the screw twist $\boldsymbol{t}$, representing the motion of the rigid body $\{b\}$ w.r.t $\{w\}$.

First, the point $\widetilde{\boldsymbol{p}}_\perp$ on the ISA closest to the origin of the reference frame is determined:

$$\widetilde{\boldsymbol{p}}_\perp = \frac{\boldsymbol{a} \times \boldsymbol{b}}{\|\boldsymbol{a}\|^2}. \tag{2.33}$$

Second, the parallel displacement $\widetilde{p}_\parallel$ along the ISA (from $\widetilde{\boldsymbol{p}}_\perp$ to $\widetilde{\boldsymbol{p}}$) is defined [34]:

$$\widetilde{p}_\parallel = \frac{\|\boldsymbol{a}\|^2}{\|\boldsymbol{a} \times \boldsymbol{a}'\|} \left(\widetilde{\boldsymbol{p}}_\perp' \cdot \boldsymbol{e}_n \right), \tag{2.34}$$

such that the *origin* of the ISA frame is found by their addition:

$$\widetilde{\boldsymbol{p}} = \widetilde{\boldsymbol{p}}_\perp + \widetilde{p}_\parallel \boldsymbol{e}_t. \tag{2.35}$$

As opposed to the FS frame, the ISA frame is a *complete* moving frame having both a position and orientation, which can be expressed using the pose matrix $\widetilde{\boldsymbol{T}}$:

$$\widetilde{\boldsymbol{T}} = \begin{bmatrix} \widetilde{\boldsymbol{R}} & \widetilde{\boldsymbol{p}} \\ \boldsymbol{0}_{3\times 1} & 1 \end{bmatrix}. \tag{2.36}$$

With the ISA frame determined, the screw invariants can be defined. The first two invariants a and b are defined, respectively, as the components of $\boldsymbol{a}$ and $\boldsymbol{b}$ along the ISA:

$$a = \boldsymbol{a} \cdot \boldsymbol{e}_t, \qquad b = \boldsymbol{b} \cdot \boldsymbol{e}_t. \tag{2.37}$$

These two invariants are referred to as *object invariants* since they relate to the object: for motion, they correspond to the rotation and translation of the

object along the ISA, while for force, they correspond to the force and moment applied to the object along the ISA.

The screw trajectory is reconstructed from the object invariants using a screw transformation matrix $\boldsymbol{S}(\boldsymbol{T})$ as in (2.7):

$$\boldsymbol{\jmath} = \boldsymbol{S}(\widetilde{\boldsymbol{T}})\ \widetilde{\boldsymbol{\jmath}}, \quad \text{with} \quad \widetilde{\boldsymbol{\jmath}} = (a\ 0\ 0\ b\ 0\ 0)^T. \tag{2.38}$$

The first-order kinematics $\widetilde{\boldsymbol{T}}'$ of the ISA frame $\widetilde{\boldsymbol{T}}$ is completely determined by four invariants:

$$\widetilde{\boldsymbol{T}}' = \widetilde{\boldsymbol{T}} \begin{bmatrix} 0 & -\omega_\kappa & 0 & v_t \\ \omega_\kappa & 0 & -\omega_\tau & 0 \\ 0 & \omega_\tau & 0 & v_b \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{2.39}$$

Here, ω_κ and ω_τ correspond to the curvature rate and torsion rate of the vector $\boldsymbol{a}$, which is the direction part of the screw $\boldsymbol{\jmath}$. Hence, the invariants ω_κ and ω_τ are calculated using similar formulas as in (2.19), with $\boldsymbol{c}$ replaced by $\boldsymbol{a}$. This is evident because, for all ξ, the orientation part $\widetilde{\boldsymbol{R}}$ in (2.36) is calculated in exactly the same way as $\widetilde{\boldsymbol{R}}$ in (2.15) for the FS frame. Hence, also its derivative is the same. The invariants v_b and v_t correspond to the translation of the origin of the ISA frame along $\boldsymbol{e}_b$ and $\boldsymbol{e}_t$ and are defined as [34]:

$$v_b = \widetilde{\boldsymbol{p}}_\perp{}' \cdot \boldsymbol{e}_b, \text{ and } v_t = \widetilde{\boldsymbol{p}}_\perp{}' \cdot \boldsymbol{e}_t + \widetilde{p}_\parallel{}'. \tag{2.40}$$

Equation (2.39) can be written compactly using (2.4):

$$\widetilde{\boldsymbol{T}}' = \widetilde{\boldsymbol{T}}\,[\boldsymbol{\imath}]_\times, \quad \text{with} \quad \boldsymbol{\imath} = (\omega_\tau\ 0\ \omega_\kappa\ v_t\ 0\ v_b)^T, \tag{2.41}$$

where $\boldsymbol{\imath}$ is referred to as the *moving frame invariant* since it models the first-order kinematics of the ISA frame. Figure 2.3 visualizes the definition of the moving frame $\widetilde{\boldsymbol{T}}$, the object invariants a and b, and the moving frame invariant $\boldsymbol{\imath}$.

Singularities: The singularities of the screw invariants bear a strong relationship with the singularities of the vector invariants. The first type of singularity occurs when the direction vector is zero ($\boldsymbol{a} = \boldsymbol{0}$). The object invariant a is zero while the moving frame invariants ω_κ, ω_τ, v_t, and v_b are undefined. In other words, the complete moving frame is arbitrary. In this case, the moment vector $\boldsymbol{b}$, i.e., $\boldsymbol{v}$ for twist and $\boldsymbol{m}$ for wrench, can still be used to construct the orientation of the moving frame using the vector invariants defined in Section 2.3. This singular case is also referred to as *infinite pitch* or *pure moment* for motion and force, respectively.

The second type of singularity occurs when the screw axis has an unchanging direction ($\boldsymbol{a} \times \boldsymbol{a}' = \boldsymbol{0}$) so that it remains parallel to itself. As a result, the invariant ω_κ is zero, while the other moving frame invariants ω_τ, v_t, and v_b are undefined. The perpendicular distance $\widetilde{\boldsymbol{p}}_\perp$ is still defined. In other words, the direction of the ISA and perpendicular distance to the ISA are well-defined in this case, but the parallel distance $\widetilde{p}_\parallel$ and normal and binormal axes are arbitrary. In this case, the derivative of the perpendicular distance $\widetilde{\boldsymbol{p}}_\perp{}'$ can be used instead of $\boldsymbol{a}$ to construct the normal and binormal axes. Thus, it is possible to determine the complete orientation and perpendicular distance of the ISA frame, but not its position along the ISA since $\widetilde{p}_\parallel$ is undefined.

Invariant properties: All the invariant properties of the vector invariants in Section 2.3 also hold for the screw invariants. The key advantage of the screw invariants compared to vector invariants is their additional invariance to the choice of the location of the reference point on the object. In other words, screw invariants are invariant to the choices of *both* the world frame $\{w\}$ and body frame $\{b\}$ in which the screw coordinates $\boldsymbol{\mathcal{s}}$ may be expressed. This property is also referred to as *bi-invariance* [76].

Trajectory reconstruction: Reconstructing the trajectory $\boldsymbol{\mathcal{s}}(\xi)$ from the invariant descriptor $(\widetilde{\boldsymbol{\mathcal{s}}}(\xi), \dot{\boldsymbol{\iota}}(\xi))$ requires numerical integration of the differential equation (2.41). Discretizing the problem again, the moving frame $\widetilde{\boldsymbol{T}}_{k+1}$ at sample $k+1$ can be found from $\widetilde{\boldsymbol{T}}_k$ at sample k as follows:

$$\widetilde{\boldsymbol{T}}_{k+1} = \widetilde{\boldsymbol{T}}_k \exp\left([\dot{\boldsymbol{\iota}}_k]_\times \Delta\xi\right), \tag{2.42}$$

where we assume that $\dot{\boldsymbol{\iota}}_k$ remains constant over the integration step $\Delta\xi$. The matrix exponential operator $\exp(\cdot)$ maps the matrix $[\dot{\boldsymbol{\iota}}]_\times \Delta\xi$ which is part of the Lie algebra $se(3)$ into the corresponding change in pose which is part of the Lie group $SE(3)$, and has a closed-form expression under the given assumptions [76]. Given the initial value for $\widetilde{\boldsymbol{T}}_1$ at $k=1$, the moving frame $\widetilde{\boldsymbol{T}}_k$ can be reconstructed for all remaining samples $k = 2...N$. From the moving frames, the corresponding screw trajectory $\boldsymbol{\mathcal{s}}_k$ is retrieved using (2.38):

$$\boldsymbol{\mathcal{s}}_k = \boldsymbol{S}(\widetilde{\boldsymbol{T}}_k)\ \widetilde{\boldsymbol{\mathcal{s}}}_k, \quad \forall k \in [1, N]. \tag{2.43}$$

Table 2.3 (bottom half) provides an overview of the screw invariants framework with applications to rigid-body twist $\boldsymbol{\mathcal{t}}$ and wrench $\boldsymbol{\mathcal{w}}$ trajectories.

2.4.2 Robust Calculation of Invariants Using Optimal Control

Calculating screw invariants $\widetilde{\boldsymbol{\mathcal{s}}}$ and $\dot{\boldsymbol{\iota}}$ with the analytical formulas has similar problems related to singularities, sensitivity to noise, and integration drift during

Table 2.3: Overview of equations for vector invariants (top half) and screw invariants (bottom half). The first column shows the type of vector and screw trajectories from which invariants can be calculated. In the second column, the local moving frame is defined in which the invariants are expressed. The third column defines the moving frame invariant that characterizes the motion of the moving frame, and the object invariant that characterizes the object's trajectory in the moving frame. The fourth column details how to reconstruct the trajectory from the invariant descriptor, possibly including the object's position and rotation.

Trajectory coordinates	**Moving frame definition**	**Invariants definition**	**Reconstruction**
Vectors $\boldsymbol{c}$: $\boldsymbol{c} = \boldsymbol{\omega}$ for rot. velocity $\boldsymbol{c} = \boldsymbol{v}$ for transl. velocity $\boldsymbol{c} = \boldsymbol{f}$ for force $\boldsymbol{c} = \boldsymbol{m}$ for moment derivatives $\boldsymbol{c}'$ and $\boldsymbol{c}''$	**orientation frame** $\widetilde{\boldsymbol{R}} = [\boldsymbol{e}_t \;\; \boldsymbol{e}_n \;\; \boldsymbol{e}_b]$, $\boldsymbol{e}_t = \dfrac{\boldsymbol{c}}{\|\boldsymbol{c}\|}$ $\boldsymbol{e}_n = \dfrac{(\boldsymbol{c} \times \boldsymbol{c}') \times \boldsymbol{c}}{\|(\boldsymbol{c} \times \boldsymbol{c}') \times \boldsymbol{c}\|}$ $\boldsymbol{e}_b = \boldsymbol{e}_t \times \boldsymbol{e}_n$	**moving frame invariant** $\boldsymbol{i} = (\omega_\tau \;\; 0 \;\; \omega_\kappa)^T$: $\omega_\kappa = \dfrac{\|\boldsymbol{c} \times \boldsymbol{c}'\|}{\|\boldsymbol{c}\|^2}$ $\omega_\tau = \dfrac{((\boldsymbol{c} \times \boldsymbol{c}') \times (\boldsymbol{c} \times \boldsymbol{c}'')) \cdot \boldsymbol{c}}{\|\boldsymbol{c} \times \boldsymbol{c}'\|^2 \|\boldsymbol{c}\|}$ **object invariant** $\widetilde{\boldsymbol{c}} = (c \;\; 0 \;\; 0)^T$ $c = \boldsymbol{c} \cdot \boldsymbol{e}_t$	**moving frame equation** $\widetilde{\boldsymbol{R}}' = \widetilde{\boldsymbol{R}}\,[\boldsymbol{i}]_\times$ **object trajectory equation** $\boldsymbol{c} = \widetilde{\boldsymbol{R}}\,\widetilde{\boldsymbol{c}}$ object orientation $\boldsymbol{R}$ equation: $\boldsymbol{R}' = [\boldsymbol{c}]_\times \boldsymbol{R}$, when $\boldsymbol{c} = \boldsymbol{\omega}$ object position $\boldsymbol{p}$ equation: $\boldsymbol{p}' = \boldsymbol{c}$, when $\boldsymbol{c} = \boldsymbol{v}$
Screws $\boldsymbol{\mathfrak{s}} = \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{pmatrix}$: $\boldsymbol{\mathfrak{s}} = \boldsymbol{t} = \begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{pmatrix}$ for twist $\boldsymbol{\mathfrak{s}} = \boldsymbol{w} = \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{m} \end{pmatrix}$ for wrench derivatives $\boldsymbol{\mathfrak{s}}'$ and $\boldsymbol{\mathfrak{s}}''$	**pose frame** $\widetilde{\boldsymbol{T}} = \begin{bmatrix} \boldsymbol{e}_t & \boldsymbol{e}_n & \boldsymbol{e}_b & \widetilde{\boldsymbol{p}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$: $\boldsymbol{e}_t = \dfrac{\boldsymbol{a}}{\|\boldsymbol{a}\|}$ $\boldsymbol{e}_n = \dfrac{(\boldsymbol{a} \times \boldsymbol{a}') \times \boldsymbol{a}}{\|(\boldsymbol{a} \times \boldsymbol{a}') \times \boldsymbol{a}\|}$ $\boldsymbol{e}_b = \boldsymbol{e}_t \times \boldsymbol{e}_n$ $\widetilde{\boldsymbol{p}} = \widetilde{\boldsymbol{p}}_\perp + \widetilde{p}_\parallel \boldsymbol{e}_t$ $\widetilde{\boldsymbol{p}}_\perp = \dfrac{\boldsymbol{a} \times \boldsymbol{b}}{\|\boldsymbol{a}\|^2}$ $\widetilde{p}_\parallel = \dfrac{\|\boldsymbol{a}\|^2}{\|\boldsymbol{a} \times \boldsymbol{a}'\|} (\widetilde{\boldsymbol{p}}_\perp{}' \cdot \boldsymbol{e}_n)$	**moving frame invariant** $\boldsymbol{\imath} = (\omega_\tau \; 0 \; \omega_\kappa \; v_t \; 0 \; v_b)^T$: $\omega_\kappa = \dfrac{\|\boldsymbol{a} \times \boldsymbol{a}'\|}{\|\boldsymbol{a}\|^2}$ $\omega_\tau = \dfrac{((\boldsymbol{a} \times \boldsymbol{a}') \times (\boldsymbol{a} \times \boldsymbol{a}'')) \cdot \boldsymbol{a}}{\|\boldsymbol{a} \times \boldsymbol{a}'\|^2 \|\boldsymbol{a}\|}$ $v_b = \widetilde{\boldsymbol{p}}_\perp{}' \cdot \boldsymbol{e}_b$ $v_t = \widetilde{\boldsymbol{p}}_\perp{}' \cdot \boldsymbol{e}_t + \widetilde{p}_\parallel{}'$ **object invariant** $\widetilde{\boldsymbol{\mathfrak{s}}} = (a \; 0 \; 0 \; b \; 0 \; 0)^T$: $a = \boldsymbol{a} \cdot \boldsymbol{e}_t$ $b = \boldsymbol{b} \cdot \boldsymbol{e}_t$	**moving frame equation** $\widetilde{\boldsymbol{T}}' = \widetilde{\boldsymbol{T}}\,[\boldsymbol{\imath}]_\times$ **object trajectory equation** $\boldsymbol{\mathfrak{s}} = \boldsymbol{S}(\widetilde{\boldsymbol{T}})\,\widetilde{\boldsymbol{\mathfrak{s}}}$ object pose $\boldsymbol{T}$ equation: $\boldsymbol{T}' = [\boldsymbol{\mathfrak{s}}]_\times \boldsymbol{T}$, when $\boldsymbol{\mathfrak{s}} = \boldsymbol{t}$

reconstruction, as mentioned in Section 2.3.2 for vector invariants.

To address these problems, we again formulate an OCP, of which the objective is to find the screw invariants that reconstruct the given measured screw trajectory in a robust way. This is reflected by introducing a cost $\Delta\boldsymbol{\mathcal{I}}_k$ that contains the difference between the reconstructed $\boldsymbol{\mathcal{I}}_k$ and measured screw trajectory $\boldsymbol{\mathcal{I}}_k^{meas}$ at sample k:

$$\Delta\boldsymbol{\mathcal{I}}_k = \boldsymbol{\mathcal{I}}_k - \boldsymbol{\mathcal{I}}_k^{meas} = \begin{pmatrix} \Delta\boldsymbol{a}_k \\ \Delta\boldsymbol{b}_k \end{pmatrix}. \tag{2.44}$$

Similarly as in Section 2.3.2, a mean-squared trajectory reconstruction error is formulated, but the errors on the directional part of the screw $\Delta\boldsymbol{a}_{\mathrm{MSE}}$ and the moment part of the screw $\Delta\boldsymbol{b}_{\mathrm{MSE}}$ are considered separately:

$$\epsilon_a^2 \geq \Delta\boldsymbol{a}_{\mathrm{MSE}}(\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{I}}^{meas}) = \frac{1}{N}\sum_{k=1}^{N} \|\Delta\boldsymbol{a}_k\|^2, \tag{2.45}$$

$$\epsilon_b^2 \geq \Delta\boldsymbol{b}_{\mathrm{MSE}}(\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{I}}^{meas}) = \frac{1}{N}\sum_{k=1}^{N} \|\Delta\boldsymbol{b}_k\|^2, \tag{2.46}$$

bounded by tolerance ϵ_a for the direction part of the screw and tolerance ϵ_b for the moment part of the screw.

A regularization cost is introduced for the moving frame invariants to achieve a more stable and smooth evolution of the moving frame $\widetilde{\boldsymbol{T}}$ in the presence of measurement noise and singularities:

$$\sum_{k=1}^{N-1} \|\dot{\boldsymbol{\iota}}_k\|_L^2 = \sum_{k=1}^{N-1} \omega_\kappa^2[k] + \omega_\tau^2[k] + \frac{1}{L^2}\left(v_n^2[k] + v_b^2[k]\right), \tag{2.47}$$

where L signifies a chosen length scale with units [m] to properly compare rotational and translational invariants.

The complete OCP to calculate the screw invariants $\widetilde{\boldsymbol{\mathcal{I}}}$ and $\dot{\boldsymbol{\iota}}$ from a given sequence of N measured screws $\boldsymbol{\mathcal{I}}_k^{meas}$ is:

$$\underset{\widetilde{\boldsymbol{\mathcal{I}}}_{[\cdot]}, \dot{\boldsymbol{\iota}}_{[\cdot]}, \boldsymbol{\mathcal{I}}_{[\cdot]}, \widetilde{\boldsymbol{T}}_{[\cdot]}}{\text{minimize}} \quad \sum_{k=1}^{N-1} \|\dot{\boldsymbol{\iota}}_k\|_L^2, \tag{2.48}$$

subject to:

$$\epsilon_a^2 \geq \Delta \boldsymbol{a}_{\mathrm{MSE}}(\boldsymbol{\mathcal{s}}, \boldsymbol{\mathcal{s}}^{meas}), \tag{r.2.45}$$

$$\epsilon_b^2 \geq \Delta \boldsymbol{b}_{\mathrm{MSE}}(\boldsymbol{\mathcal{s}}, \boldsymbol{\mathcal{s}}^{meas}), \tag{r.2.46}$$

$$\widetilde{\boldsymbol{T}}_{k+1} = \widetilde{\boldsymbol{T}}_k \exp\left([\boldsymbol{\dot{\iota}}_k]_\times \Delta\xi\right), \quad \forall k \in [1, N-1] \tag{r.2.42}$$

$$\boldsymbol{\mathcal{s}}_k = \boldsymbol{S}(\widetilde{\boldsymbol{T}}_k)\ \widetilde{\boldsymbol{\mathcal{s}}}_k, \qquad \forall k \in [1, N] \tag{r.2.43}$$

$$\widetilde{\boldsymbol{T}}_1^{-1} \widetilde{\boldsymbol{T}}_1 = \boldsymbol{I}_{4\times 4}. \tag{2.49}$$

Besides the trajectory error constraints (2.45)-(2.46), the constraints consist of the trajectory reconstruction equations (2.42)-(2.43), supplemented with an orthonormality constraint on the initial moving frame $\widetilde{\boldsymbol{T}}_1$.[6]

Initialization: Similarly as for vector invariants, an invariant initialization of the variables in the OCP can be done using either the analytical formulas or an approach based on an average moving frame [10]. The latter is summarized as follows. The orientation of the moving frames $\widetilde{\boldsymbol{T}}_k$ is initialized using the same approach as for vector trajectories, based on the directional part of the measured screw trajectory $\boldsymbol{\mathcal{s}}_k^{meas}$. The origin of the moving frames $\widetilde{\boldsymbol{T}}_k$ is initialized using the average intersection point of all the instantaneous screw axes corresponding to the measured screw trajectory $\boldsymbol{\mathcal{s}}^{meas}$ [10]. The moving frame invariants ω_κ, ω_τ, v_b, and v_t are initialized with zero values. The object invariants a_k and b_k are initialized by transforming the screw trajectory $\boldsymbol{\mathcal{s}}_k^{meas}$ to the average moving frame and selecting the components along the first axis for each sample k.

Application to motion and force data: The calculation of screw invariants using OCP (2.48)-(2.49) can directly be applied to twist trajectories $\boldsymbol{\mathcal{t}}(\xi)$ and wrench trajectories $\boldsymbol{\mathcal{w}}(\xi)$.

Similarly as in Section 2.3.2, if the measured motion data consists of positions and orientations of the rigid body, instead of twists, the optimization problem (2.48)-(2.49) can be adapted to include the object's pose. Below, this is worked out assuming the positions and orientations are given by homogeneous transformation matrices. The proposed approach can be adapted to other pose representations, such as dual quaternions.

Including pose measurements for case $\boldsymbol{\mathcal{s}} = \boldsymbol{\mathcal{t}}$: Additional variables are introduced for the reconstructed pose matrices $\boldsymbol{T}_k = \begin{bmatrix} \boldsymbol{R}_k & \boldsymbol{p}_k \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix}$ at each sample

[6] Similar to (2.26), implementing this constraint in practice remains an open challenge. In the current implementation, it is applied as a constraint on the rotation matrix component of $\widetilde{\boldsymbol{T}}_1$, using either the full or triangular constraint formulations.

k. The trajectory reconstruction costs $\Delta \boldsymbol{a}_{\mathrm{MSE}}$ and $\Delta \boldsymbol{b}_{\mathrm{MSE}}$ in (2.45)-(2.46) are replaced by the MSE errors between the reconstructed and the measured rotation matrices and positions:

$$\Delta \boldsymbol{R}_{\mathrm{MSE}}(\boldsymbol{R}, \boldsymbol{R}^{meas}) = \frac{1}{N} \sum_{k=1}^{N} \left\| (\boldsymbol{R}_k^{meas})^T \boldsymbol{R}_k - \boldsymbol{I}_{3\times 3} \right\|_{F_u}^2, \tag{2.50}$$

$$\Delta \boldsymbol{p}_{\mathrm{MSE}}(\boldsymbol{p}, \boldsymbol{p}^{meas}) = \frac{1}{N} \sum_{k=1}^{N} \| \boldsymbol{p}_k - \boldsymbol{p}_k^{meas} \|^2. \tag{2.51}$$

The reconstructed pose matrices $\boldsymbol{T}_k$ are defined by integrating $\boldsymbol{\jmath}_k$, in this case signifying the object's twist, in additional reconstruction constraints:

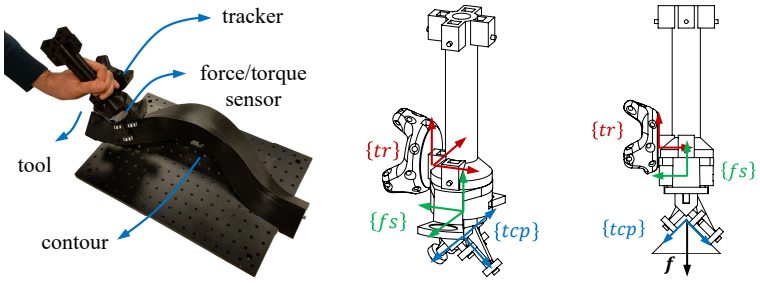
$$\boldsymbol{T}_{k+1} = \exp\left([\boldsymbol{\jmath}_k]_\times \, \Delta\xi \right) \boldsymbol{T}_k, \quad \forall k \in [0, N]. \tag{2.52}$$

Finally, adding an orthonormality constraint on the initial pose $\boldsymbol{T}_1$, like in (2.49), ensures that orthonormality is preserved.[7]

2.5 Application to 3D Contour Following

This section demonstrates the use and benefits of screw and vector invariant descriptors for both motion and force trajectories by analyzing a human-demonstrated 3D contour following task, shown in Figure 2.4. The human operator holds the tool while following the contour and pressing the tool's five contact wheels (diameter 22 mm) against the contour.

[7] The same considerations discussed for (2.49) also apply in this case.



(a) Setup (b) Frames (c) Cross-section

Figure 2.4: The 3D contour following setup: (a) demonstration setup showing tool and contour, (b) 3D drawing showing the three assigned frames on the tool: tracker $\{tr\}$, force/torque sensor $\{fs\}$, and tool center point (TCP) $\{tcp\}$, (c) cross-section of the contour showing the reference contact force $\boldsymbol{f}$.

The contour consists of the edge between two contact surfaces that form an angle of 90° in each point along the contour (see Figure 2.4c). Each of the five contact wheels takes away one degree of freedom for the motion of the tool relative to the contour, while creating one degree of freedom to apply a contact force. Hence, every point along the contour is characterized by a 1-DOF vector space of possible twists and a 5-DOF vector space of possible contact wrenches. This makes the application interesting, as we expect to obtain a very good repeatability of the invariant descriptors for motion among different demonstrations, while we expect to see the effect of human variation among the demonstrations in the invariant descriptors for force. Furthermore, the contour is designed to have a symmetric curvature profile and an antisymmetric torsion profile, both with respect to its midpoint. The contour starts and ends with a straight line segment (no curvature or torsion).

2.5.1 Experimental Set-up

An HTC VIVE motion capture system, consisting of a tracker mounted on the tool and a camera (not shown in Figure 2.4), records the pose trajectory of the tool with respect to a reference frame attached to the camera. This frame is chosen as the world frame $\{w\}$. The system records the pose trajectory with a frequency of 200 Hz and an accuracy in the order of a few millimeters and a few degrees. The contact force/torque is measured with the same frequency using a 6-axis JR3 force/torque sensor. The nominal accuracy of this force/torque sensor is $\pm 1\%$ of its standard measurement range which is ± 800 N along the

central axis (Z-axis) and ±400 N in the horizontal plane (XY-plane) for the force, and ±24 Nm in all directions for the moment.

Three frames are rigidly attached to the tool: the motion tracker frame $\{tr\}$, the force/torque sensor frame $\{fs\}$, and the *tool-center-point* (TCP) frame $\{tcp\}$. The $\{tcp\}$ frame is defined such that, when the tool is tracking a straight-line segment, the origin of $\{tcp\}$ corresponds to the intersection of the contour's edge and the symmetry axis of the cylindrical part of the tool (see Figure 2.4c). One axis of the $\{tcp\}$ frame is along the edge and the remaining two axes are along the two surfaces. (For completeness: if the edge has curvature and/or torsion, the $\{tcp\}$ frame will deviate slightly from this contour-centered definition due to the complex contact geometry).

To introduce variation in the measurements, the tracker can be physically attached to the tool at different locations, further discussed in Section 2.5.3. However, the location of the force sensor relative to the tool cannot be changed in the current set-up. Instead, and without loss of generality, we can artificially transform the physically measured wrenches to a different, virtual frame on the tool using a screw transformation matrix $\boldsymbol{S}$ as in (2.9). We can take this one step further and in a similar manner transform the physically measured wrenches to the world frame $\{w\}$. This way we can, again without loss of generality, simulate an alternative experimental set-up in which the force sensor is fixed to the world instead of the tool, i.e., representing a situation where the force sensor is placed underneath the support to which the contour is fixed.

2.5.2 Objectives

A first objective is to show that the invariant descriptors have a physical interpretation and to obtain evidence that they are calculated correctly using the OCPs. It is however very difficult to calculate the ground truth invariants given the complex contact geometry involving five contact wheels with finite radii and finite mutual distances. Instead, we calculated reference values to compare the invariants with, for the limit case of an infinitely small set of contact wheels and a contact wrench which corresponds to a pure force with constant magnitude $\|\boldsymbol{f}\| = 25$ N, as shown in Figure 2.4c. In this limit case, the pose of $\{tcp\}$ is easily derived from the contour's CAD data. The wrench is then also uniquely defined and constant in any frame attached to the tool. From the reference pose and reference wrench we calculate the corresponding invariant descriptors using the same OCPs as for the human-demonstrated data.

A second objective is to confirm the invariant properties of the vector invariants and the screw invariants, as discussed in Sections 2.3 and 2.4, respectively. Furthermore, we want to point out the practical benefits of these properties in

terms of relaxed calibration needs. We also want to show the difference between the invariant descriptors depending on the perspective of the input data: as the motion and wrench trajectories can be defined either with respect to the world or with respect to the moving body, the corresponding moving frame invariants will also model the motion of the moving frame with respect to the world or the moving body, and hence they will be different.

A third objective is to confirm that the new formulation of the OCPs for calculating the invariants, as introduced in Sections 2.3.2 and 2.4.2, allows for an intuitive and effective parameter tuning.

A fourth and final objective is to confirm that the trajectories reconstructed from the calculated invariants are accurate and driftless, even when the trajectories are reconstructed in a different region in space.

2.5.3 Experiment Design

Twelve demonstrations of the 3D contour following task were recorded. According to the experimental protocol, the operator was asked to repeat the task twelve times by starting from approximately the same location (at the end of the initial straight-line segment) and approximately ending at the same location (at the beginning of the final straight-line segment). The operator was also asked to try to maintain a constant pure contact force as shown in Figure 2.4c, because this can be shown to be a very good strategy to ensure that the five wheels remain in contact with the contour. Both instructions contributed to limiting human variation in the task execution.

Each of the twelve demonstrations was recorded with the tracker physically attached to the tool in one of six different locations, shown in Figure 2.5 (i.e. two demonstrations per tracker location), resulting in significantly different pose trajectories.

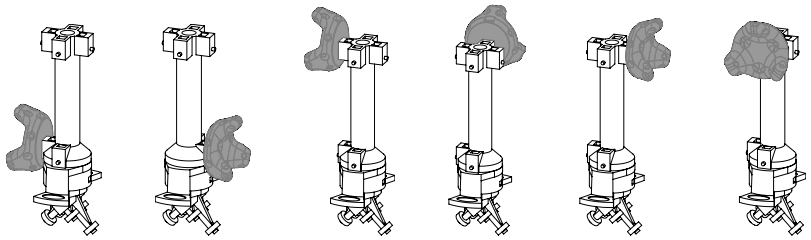


Figure 2.5: The six different configurations of the motion tracker on the tool.

Additionally, the physically measured contact wrenches were artificially transformed from the force sensor frame $\{fs\}$ to the tracker frame $\{tr\}$ used in the demonstration, simulating a change in location of the force sensor and resulting in significantly different wrench trajectories.

The screw invariants are expected to be invariant for the previous variations, but the vector invariants are not. To verify the repeatability of the vector invariants, it is necessary to pick a well-chosen reference point on the tool to express the position and moment trajectories, the same reference point for all demonstrations. The obvious choice was the origin of $\{tcp\}$, both for the position and moment trajectories. Hence, for this specific case, the measured pose and wrench trajectories of the twelve demonstrations were additionally transformed to the $\{tcp\}$ frame. Note that this requires calibration of the origins of $\{tr\}$ and $\{fs\}$ with respect to $\{tcp\}$, but this is required solely for the vector invariants, not the screw invariants.

The twelve resulting pose trajectories were further modified by changing the pose of the world reference frame relative to the contour, simulating a change in camera viewpoint. Table 2.4 lists the different artificial transformations that were applied to each of the twelve trials together with their corresponding tracker configuration. Changing the world reference frame is not expected to affect the screw or vector invariants for motion. Obviously, such change also does not affect the wrench trajectories and the resulting wrench invariants since the force sensor is attached to the tool, not to the world.

To simulate the alternative experimental set-up in which the force sensor is fixed to the world, the twelve measured wrench trajectories were artificially transformed to a single frame, fixed to the world underneath the contour, about halfway the contour. The exact location of this frame relative to the contour is not known or calibrated, but this does not affect the screw invariants.

Table 2.4: Each of the twelve demonstration trials is listed with one of the six configurations of the motion tracker (see Figure 2.5) and with the artificial transformation that was applied to the measured poses to simulate a large variation in the world (i.e. camera) frame. These transformations are specified using XYZ Euler angles and a translation vector.

Trial # +config.	**Artificial transformation** angles [°]	transl. [m]	**Trial # +config.**	**Artificial transformation** angles [°]	transl. [m]
1 (a)	[0, -60, 0]	[-3, 2, -1]	7 (d)	[-90, 0, 45]	[1, 0.5, -0.5]
2 (a)	[30, 0, -180]	[-1, 2.5, -1]	8 (d)	[0, 90, -135]	[-1, 0, -0.5]
3 (b)	[0, -60, -180]	[-1, 2, -2]	9 (e)	[60, 0, -150]	[1, -2, 2]
4 (b)	[45, 0, 0]	[-3, 2.5, -2]	10 (e)	[0, 45, -60]	[3, -2, 2]
5 (c)	[-90, 45, 90]	[-1, 0.5, 0.5]	11 (f)	[0, -45, -180]	[3, -1.5, 1]
6 (c)	[0, 90, 45]	[1, 0, 0.5]	12 (f)	[-60, 0, -45]	[1, -1.5, 1]

Finally, to show that the trajectories reconstructed from the invariants are accurate and driftless, the motion and force trajectories were reconstructed at a new location and were compared with the measured trajectories, after they were globally transformed to the same location.

2.5.4 Data Processing

This section explains how the screw and vector invariants are calculated from the measured pose and wrench trajectories and the reference pose and wrench trajectories.

Preprocessing: There are two preprocessing steps: segmentation and re-parameterization.

The purpose of segmentation is to extract the motion-in-contact part from a complete demonstration, which also includes an approach-to-contact motion, a retract motion, and two stationary parts while in contact, at the start and the end of the actual contour tracking motion. Segmentation is achieved by using a threshold for the magnitude of the force (5 N) to detect contact, and a threshold for the magnitude of the translational velocity of the TCP (0.05 m/s) to remove the stationary parts of the demonstration.

Re-parameterization involves changing the time-based input trajectories to trajectories that are function of an alternative progress variable. As discussed in Section 2.2.3, for tasks with dominant translation like contour following, the integral of the translational velocity is a natural choice. Accordingly, we chose the progress variable ξ as the integral of the TCP's velocity, because it best reflects the arc length along the contour. Finally, we rescaled the progress

variable to a dimensionless variable ranging from 0 to 1, making it independent of the scale of the contour.

Calculation of the invariants: The force invariants were calculated from the wrench measurements using the original OCPs: (2.25)-(2.26) for vector invariants and (2.48)-(2.49) for screw invariants. The motion invariants were calculated from the pose measurements using the adapted versions of the OCPs using (2.28)-(2.31) for the vector invariants and (2.50)-(2.52) for the screw invariants. All OCPs were specified using the nonlinear optimization framework CasADi [11] and solved using a primal-dual Newton method with IPOPT [110].

Initialization of the OCP: For initializing the moving frames, we followed the initialization approaches based on an average moving frame as explained in Sections 2.3.2 and 2.4.2 with a minor variation. Due to the symmetry and antisymmetry in the contour, averages calculated over the trajectory might become approximately zero. To avoid confusion in determining the signs, we calculated the average over the last two-thirds of the trajectory. The motion invariants are calculated using pose measurements as input. However, the initialization routine uses velocity and twist data as input. Hence, for motion, numerical differentiation is performed to estimate the velocities and twists as inputs for the initialization routine.

Tuning of the OCPs: Vector and screw invariants for motion and force were calculated using measured position, orientation, force, and moment trajectories as inputs. For every type of trajectory, a tolerance ϵ related to the desired accuracy of the reconstructed trajectory had to be chosen, such as for example in (2.24) for the vector trajectories. To maintain an intuitive relation between the values for these tolerances and the accuracy of the measurement systems, we expressed these tolerances in the respective sensor frames. All reported results were obtained using the following intuitive values: $\epsilon_p = 2$ mm for position and $\epsilon_R = 2^\circ$ for orientation, which corresponded to a rough estimation of the accuracy of the motion capture system, and $\epsilon_f = 0.8$ N for force and $\epsilon_m = 0.16$ Nm for moment, which corresponded to the estimated noise levels of the force/torque measurements. To calculate screw invariants, a length scale L must be chosen to properly compare rotational and translational invariants. For the application we chose $L = 0.5$ m, i.e. the length of the contour's edge.

Reconstructing trajectories from invariants: The pose trajectory $\boldsymbol{T}(\xi)$ of the tool and the contact wrench trajectory $\boldsymbol{w}(\xi)$ were reconstructed from the screw invariant descriptors at a new location in space. This was done by supplying new initial moving frames $\widetilde{\boldsymbol{T}}(0)$ for both motion and wrench, and a new initial pose of the tool $\boldsymbol{T}(0)$ in the trajectory reconstruction equations (2.42), (2.43) and (2.52). The frames were specified by globally transforming the original calculated initial frames to the new location. Time-based trajectories

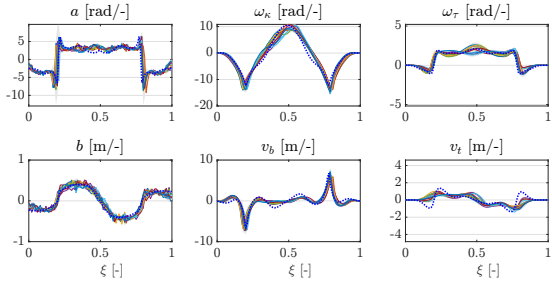
$\boldsymbol{T}(t)$ and $\boldsymbol{\omega}(t)$ were obtained by re-applying the time profile $\xi(t)$ that was extracted from the demonstration using the inverse of the reparameterization in (2.11).

2.5.5 Results and Discussion

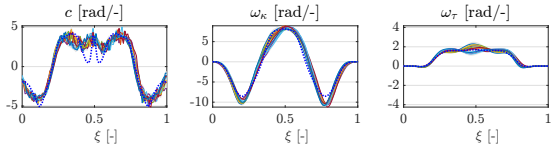
Figures 2.6-2.11 summarize the experimental results. Figures 2.6-2.8 contain screw and vector invariants: motion invariants with respect to the world (Figure 2.6), force invariants with respect to the tool (Figure 2.7), and force invariants with respect to the world (Figure 2.8). Additional results for other cases can be generated in the provided software [7]. Each plot contains twelve thin lines corresponding to each of the twelve demonstrations. The thick dotted blue line represents the reference values for the limit case. The variation among the demonstrations is given by the two-sigma band depicted in gray. For all invariants, the dimensionless arc length ξ defined in Section 2.5.4 is chosen as the progress variable. Figure 2.10 visualizes the corresponding evolution of the different moving frames for one demonstration. Note that in the subfigures of Figure 2.10, the contour is only positioned approximately[8] because its true pose was not recorded. Figure 2.11 visualizes motion and force trajectories reconstructed from the invariants at a new location. Below, the results are discussed in detail.

[8]This approximation is based on the spatial alignment of the recorded trajectory of $\{tcp\}$ with the edge of the contour [91].

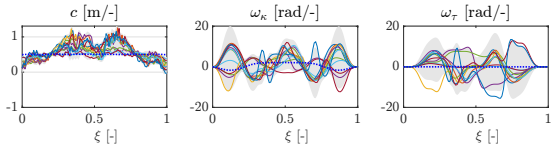
Motion invariants with respect to world:



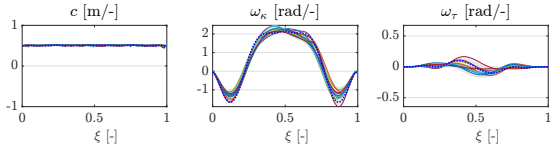
(a) Screw invariants calculated from measured pose of tracker



(b) Vector invariants calculated from measured orientation of tracker



(c) Vector invariants calculated from measured position of tracker



(d) Vector invariants calculated from estimated position of tool point

Figure 2.6: Screw and vector invariants calculated from position and orientation of tracker or tool point on manipulated tool, measured in the world frame.

Figure 2.6a shows the screw invariants for motion. The object invariants a and b describe the rotation and translation of the object along the screw axis, while the moving frame invariants ω_κ, ω_τ, v_b, and v_t describe the rotation and translation of the moving frame attached to the screw axis. Recalling that

each demonstration has different tracker and world frames, the relatively small variations between demonstrations confirm the invariance properties of the screw invariants. Even in the presence of human variations, relatively small variations between the demonstrations and good correspondence with the reference values from the limit case are obtained. This is explained by the constrained motion, allowing variations in only one DOF. Due to the special design of the contour, clear symmetric or antisymmetric profiles with respect to the midpoint were expected and were obtained. The motion of the moving frames for one trial is visualized in Figure 2.10a.

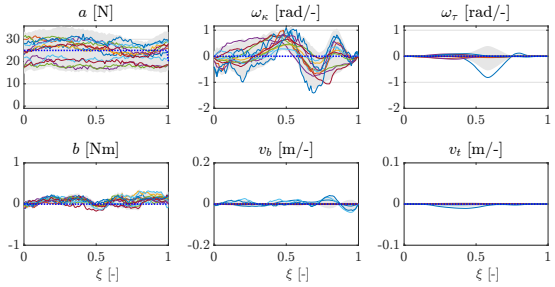
Figure 2.6b shows the vector invariants for orientation. According to the analytical formulas, they should correspond to the invariants for the directional component of the screw for motion (top row in Figure 2.6a). The reason why they are not exactly the same is explained by the OCP-based calculation approach. In this approach, the calculation for orientation and translation are coupled into a single OCP for the screw invariants and hence they are subject to a joint regularization. Nevertheless, the overall evolution of the vector invariants for orientation and the invariants for the directional component of the screw are still similar.

Turning to the vector invariants for translation, Figure 2.6c confirms that they are not invariant for changes of the tracker location, serving as the reference point to define the translation of the tool. This non-invariance can be resolved by picking the same reference point for all demonstrations, in this case the origin of $\{tcp\}$, as shown in Figure 2.6d. In this figure, c represents the magnitude of the translational velocity vector, while ω_κ and ω_τ represent the dimensionless curvature and torsion of the point curve tracked by the origin of $\{tcp\}$. From all the figures, Figure 2.6d exhibits the best correspondence between the reference limit case and the demonstrations. This is to be expected since the limit case models the translation of the $\{tcp\}$-frame along the edge. For this limit case, c is approximately constant (about 0.5 m/-). Its integral (also close to 0.5 m) represents the arc length of the point curve tracked by the origin of $\{tcp\}$. The small values for the torsion rate indicate that this point curve lies, at least locally, approximately in a plane. The small differences between the limit case and the demonstrations are due to the finite dimensions of the tracking tool and due to measurement errors. The moving FS frames corresponding to these vector invariants for one demonstration are visualized in Figure 2.10d.

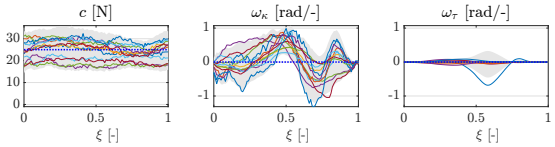
Given the geometry-based definition of the invariants, they can be used to segment the trajectory into meaningful parts. For example, in the screw invariants in Figure 2.6a, we notice two inflection points at $\xi \approx 0.2$ and at $\xi \approx 0.8$: while the rotational velocity a changes sign at these progress values, the peaks in v_b show that the position of the ISA is rapidly shifting from a center of curvature on one side of the contour to the other side of the contour.

This is confirmed by the visualization of the corresponding ISAs in Figure 2.10a.

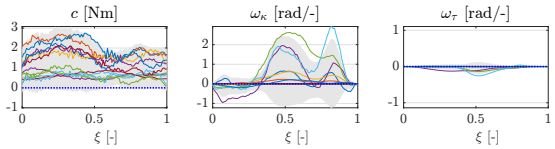
Force invariants with respect to tool:



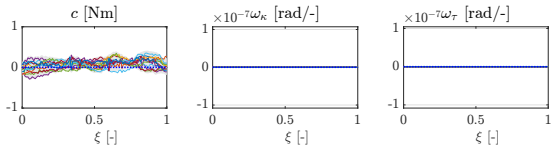
(a) Screw invariants calculated from wrench transformed to tracker frame



(b) Vector invariants calculated from force transformed to tracker frame



(c) Vector invariants calculated from moment transformed to tracker frame



(d) Vector invariants calculated from moment transformed to tool center frame

Figure 2.7: Screw and vector invariants calculated from measured force and moment trajectories, transformed to either the tracker or the tool center frame.

More variation among demonstrations was expected in the wrench invariants because, although the operator was instructed to apply a pure force in a specified

direction and through a specific point of the tool, the operator could at any time apply a wrench in a local 5-DOF space. The higher noise levels associated with force measurements may cause additional variation compared to the motion invariants. Note that a pure force that remains parallel to itself corresponds to the second type of singularity, mentioned in Section 2.4. This singularity applies to both vector and screw invariants.

Figure 2.7a shows the screw invariants. Recalling that each demonstration has a different transformed force sensor frame, these plots confirm that the screw invariants are invariant for such transformation. The plot of invariant a shows that the operator was able to maintain relatively constant magnitudes of the force along the contour, although the magnitudes varied between demonstrations. The plots of ω_κ and ω_τ reveal a limited change of direction of the force (hence of the ISA), almost in a single plane (ω_τ is very small). This plane is orthogonal to the edge of the contour. This can be found by analyzing the direction of the ISAs and is confirmed in Figure 2.10b, where the first (red) and second axes (green) of the successive moving frames lie approximately in a plane that is orthogonal to the tangent of the contour.

The second row of Figure 2.7a shows that there is no significant moment about the ISA (b is very small) and that the origin of the moving frame does not translate much, both parallel to and along the ISA (both v_b and v_t remain very small). The latter can also be seen in Figure 2.10b.

Figure 2.7b shows the vector invariants for force, which are nearly identical to the directional component of the screw invariants (top row in Figure 2.7a). As for the vector invariants for moment, Figure 2.7c confirms that they are not invariant for a change in position of the force sensor frame, resulting in moment magnitudes up to 2.5 Nm. Similarly as for translational velocity, this can be resolved by picking the TCP as the common reference point for expressing the moments in all demonstrations, as shown in Figure 2.7d. Since these moment values are relatively close to zero with respect to the specified tolerance $\epsilon_m = 0.16$ Nm, they can be perfectly described by a stationary Frenet-Serret frame. Therefore, the solutions of ω_κ and ω_τ are below 10^{-8}, the stopping criterion in the OCP solver.

Force invariants with respect to world:

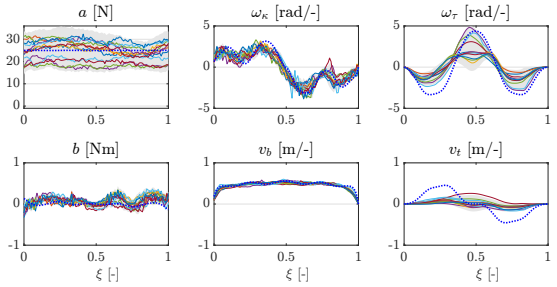


Figure 2.8: Screw invariants calculated from measured wrench transformed to the world frame.

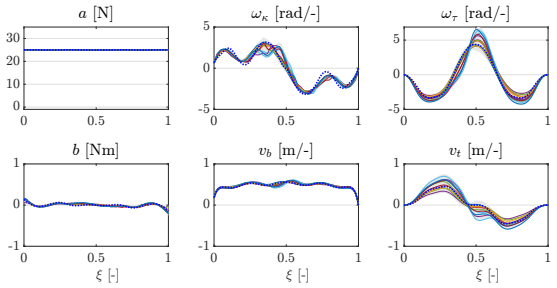
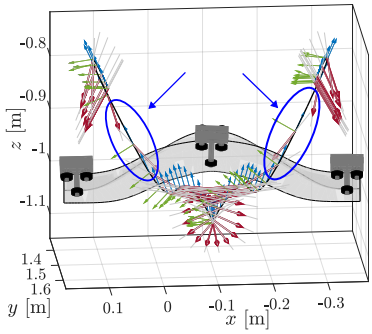


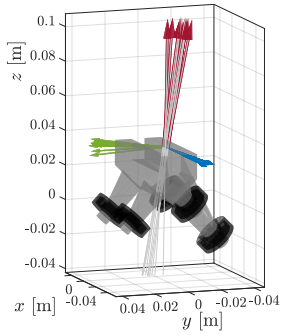
Figure 2.9: Screw invariants calculated from a *simulated wrench* in the world frame to study the reduction in variation with respect to Figure 2.8.

Figure 2.8 shows the screw invariants for the simulated, alternative experimental set-up in which the force sensor is assumed fixed to the world. This simulation, in particular the transformation of the physical force measurements to the world, is subject to two additional errors: the calibration between the force sensor and the motion tracker, and the camera measurements. Again, the invariants are shown to be insensitive to this transformation. Figure 2.10c shows the evolution of the corresponding moving frame. During execution, the force ISA remains approximately perpendicular to the contour edge while it approximately intersects the contour edge. To do this for the entire trajectory, the force axis has to translate along the contour with a velocity v_b, which is approximately equal to the translation of the TCP per dimensionless arc length (about 0.5 m/-). Integrating this value over the total horizon again results in the length of the entire contour ($\sim$0.5 m).

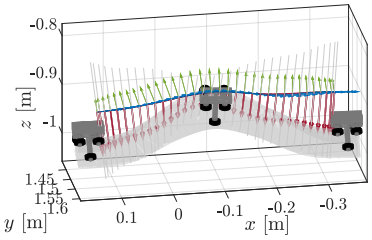
To further study the variation in the force invariants with respect to the world, we replaced the real wrench measurements in each demonstration with a constant simulated wrench $\boldsymbol{w} = 25\begin{bmatrix}0 & \cos(\frac{\pi}{4}) & \sin(\frac{\pi}{4}) & 0 & 0 & 0\end{bmatrix}^T$ expressed in the TCP frame. By using this simulated wrench, the effects of human variations and wrench measurement noise were eliminated. Consequently, the corresponding force invariants with respect to the world are only influenced by errors in the calibration between the TCP and the motion tracker, and noise in the motion measurements. Invariant descriptors of this case are shown in Figure 2.9. We observe greater repeatability in force invariants compared to Figure 2.8 as a result of eliminating the human variation and wrench measurement noise. This analysis confirms that the larger variation in the force invariants compared to motion invariants is mainly due to human variation in a 5-DOF space, not due to errors in the computation.



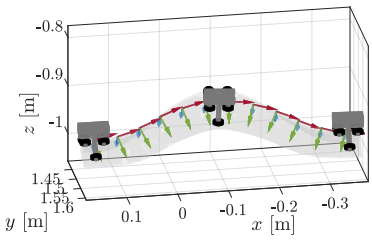
(a) Motion ISA frames in world



(b) Force ISA frames in tool



(c) Force ISA frames in world



(d) Transl. velocity FS frames in world

Figure 2.10: Moving frames calculated from data of Trial 5: (a) ISA frames calculated from the tool's motion in the world. The red arrows correspond to the ISA itself, which is the first axis of the moving frame. The green and blue arrows correspond to the second and third axes, respectively. The two blue circles indicate the translation of the ISA frames towards the other side of the contour at the inflection points; (b) ISA frames calculated from interaction wrench, relative to the tool; (c) ISA frames calculated from wrench w.r.t the world; (d) Frenet-Serret (FS) frames calculated from the position of the tool center point (TCP). The red arrows correspond to the tangent along the contour, which is the first axis of the moving frame.

Reconstructed motion and force trajectories: Figure 2.11 visualizes a reconstruction of the motion and force trajectory at a new location starting from the invariants of one of the trials. Table 2.5 compares these reconstructed trajectories with the measured trajectories on which a global transformation has been applied so that they are in the same location. The RMS-differences between the trajectories were found to be in agreement with the chosen tolerances for the trajectory accuracy constraint in the OCPs. The small difference is due to the convergence tolerance of the numerical solver of the OCPs, which was

set to 10^{-8}. These results confirm that the trajectories reconstructed from the calculated invariants have the accuracy specified in the OCPs and are also driftless. Evidently, these results also hold for the special case of a reconstruction at the same location as the measurements.

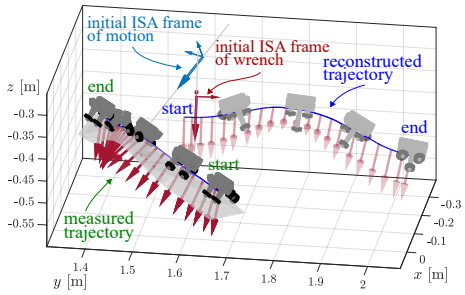


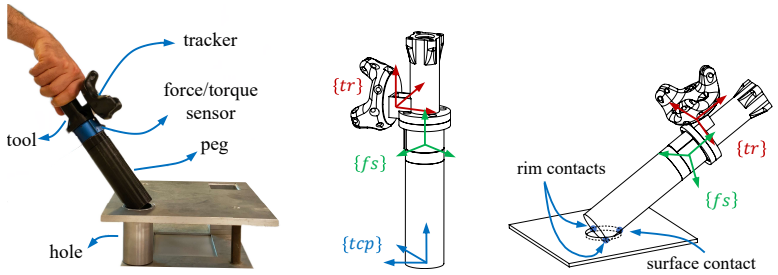
Figure 2.11: Reconstruction of the motion and wrench trajectory of Trial 7 at a new location given the screw invariants of the measured trajectory seen from the world. The motion of the follower is shown using the position of the TCP and the orientation of the follower. The force on the follower is depicted by the red arrows at the corresponding position of the TCP. The initial ISA frames for motion and wrench, necessary for reconstruction, are also depicted.

Table 2.5: Reconstruction error for the motion and force trajectory of Figure 2.11. The chosen tolerances are given as a reference.

Trajectory	**RMS reconstruction error**	**Tolerance in OCP**
position [mm]	$2.00 + 7 \times 10^{-9}$	$\epsilon_p = 2.00$
orientation [°]	$2.00 + 9 \times 10^{-9}$	$\epsilon_R = 2.00$
force [N]	$0.80 + 4 \times 10^{-9}$	$\epsilon_f = 0.80$
moment [Nm]	$0.16 + 8 \times 10^{-9}$	$\epsilon_m = 0.16$

2.6 Application to Peg-on-Hole Alignment

This section examines motion and force trajectories resulting from a human-demonstrated peg-on-hole task in which the operator holds a tool to which the peg with diameter 50 mm is attached, as shown in Figure 2.12.



(a) Setup (b) Frames (c) Three-point contact

Figure 2.12: Peg-on-hole setup with defined frames and illustration of the contact.

The task aims to ensure that peg and hole are aligned before peg insertion starts, even if the hole is inaccurately located. To accomplish this, the operator first establishes a stable three-point contact between the hole and a purposefully misaligned peg, as shown in Figure 2.12c. This contact involves two rim-rim contact points and one surface-rim contact point. We recorded 12 demonstrations of the alignment, each starting from such initial three-point contact and finishing at full alignment. The operator was instructed to perform alignment while maintaining the three-point contact. All demonstrations were performed by the same operator, but in two batches, respectively of 7 and of 5 trials, at different points in time.

Although this task appears to involve a pure rotation, previous theoretical studies have shown that, apart from a rotation, the motion involves a small, but non-zero translation, see e.g. [25]. It is evident that, due to the three-point contact and neglecting friction, this task involves two instantaneous 3-DOF vector spaces: one for the wrench and one for the motion. So, the operator can use different motion and wrench strategies during the demonstration.

2.6.1 Experimental Set-up and Data Processing

The motion capture system is the same as for the contour following task of Section 2.5. The force/torque sensor is again a 6-axis JR3 sensor with a nominal accuracy of $\pm 1\%$, but its measurement ranges are smaller: ± 200 N along Z and ± 100 N along X and Y for force, and ± 5 Nm in all directions for moment. The same three frames as on the contour following tool were defined on the demonstration tool here (see Figure 2.12b), with the TCP frame located at the tip of the peg.

Unlike in the contour following application, we did not use different tracker positions on the tool and did not artificially transform the pose measurements by changing the world frame, because invariance with respect to such variations was sufficiently proven in the contour following experiment. We also did not transform the wrench measurements to different locations on the tool, but we did transform them to a frame fixed to the world to investigate the wrench invariants with respect to the world.

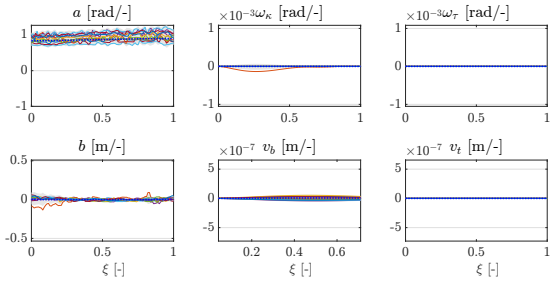
Data was processed in roughly the same way as in the contour following application, with the following changes or additions. The segmentation thresholds were set to 1 N and 0.35 rad/s for force and angular velocity, respectively. Since the task is predominantly rotational, the integral of the rotational velocity was chosen as the progress variable to re-parameterize the time-based input trajectories, and it was also rescaled to a dimensionless variable ranging from 0 to 1. The initialization of the OCPs based on average moving frames (Section 2.3.2 and 2.4.2) was done on the complete measured trajectories. Furthermore, since the force/torque sensor was more accurate than the one used for contour following (same relative accuracy for a smaller range), we lowered tolerances ϵ_f and ϵ_m in the OCPs to 0.3 N and 0.1 Nm, respectively.

Finally, to investigate if the operator used different wrench strategies in the two batches, we calculated the invariants corresponding to the average of the wrench trajectories for each of the two batches and compared them to the invariants corresponding to the average of all demonstrations. This was done both with respect to the tool and with respect to the world. Three remarks have to made here: 1) we took the average of the wrench trajectories rather than the average of the invariants, because averaging is a linear operation, but the subsequent calculation of the invariants is not; 2) we can just average the measured wrench trajectories, because all wrench trajectories were recorded at the same physical sensor location on the tool and were additionally transformed to the same virtual location in the world[9]; 3) because averaged wrench trajectories are less noisy, we could further lower ϵ_f to 0.17 N and ϵ_m to 0.05 Nm.

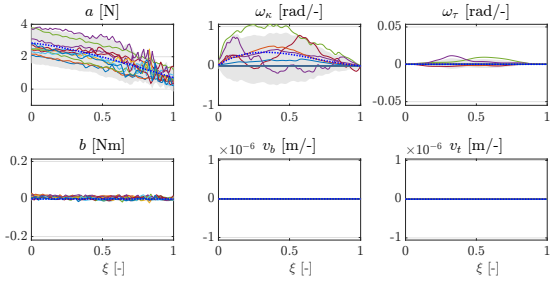
2.6.2 Results and Discussion

Figure 2.13 and 2.14 summarize the experimental results, focusing on screw invariants for motion and force. Additional results for the vector invariants can be generated in the provided software [7].

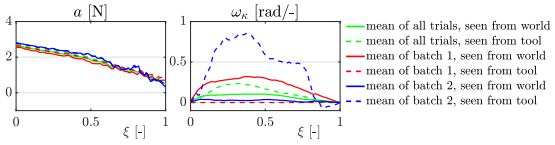
[9]If the wrench trajectories are recorded at different locations, we can always calculate the invariants of each wrench trajectory and then reconstruct all wrench trajectories at a common new location, as explained in Section 2.5.4.



(a) Screw invariants for motion with respect to world

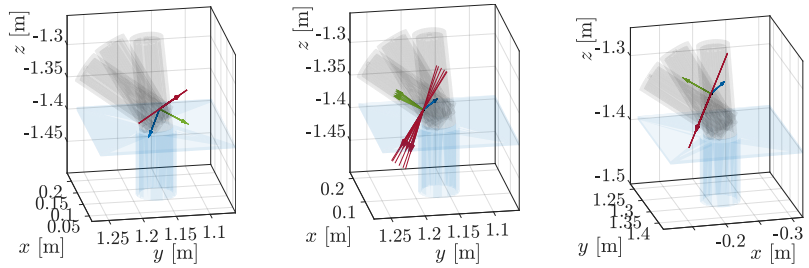


(b) Screw invariants for force with respect to world



(c) First two screw invariants for the averaged wrench trajectories of measurement batch 1 and 2, with respect to either world or tool.

Figure 2.13: Screw invariants for motion and force for the peg-on-hole task.



(a) Motion ISA Trial 1 (b) Force ISA Trial 1 (c) Force ISA Trial 12

Figure 2.14: Moving frames for trials in the peg-on-hole alignment task, all seen from the world frame. The ISAs are depicted using red arrows: (a) Moving frames calculated from the tool's motion for Trial 1 (batch 1), showing a stationary ISA; (b) Moving frames calculated from the wrench for Trial 1 (batch 1), showing a moving ISA; (c) Moving frames of the wrench for Trial 12 (batch 2), showing a stationary ISA.

Motion invariants with respect to world: Figure 2.13a shows the screw invariants of motion. The object invariant a corresponds to the relative rotation of the peg with respect to the hole around the screw axis. Its value is almost constant because the re-parameterization is based on the rotation and it lies around 1 rad/-. This makes sense since the integral of a corresponds to the initial misalignment angle, which is about 1 rad. Object invariant b is small, meaning there is almost no translation along the screw axis. The moving frame invariants ω_κ, ω_τ, v_b, and v_t are zero or near zero, indicating that the moving frame is stationary. Hence, with the chosen values of the tolerances in the OCP, the relative motion between peg and hole can be explained by a pure rotation about a fixed axis. As a result, the small theoretical translation mentioned earlier vanishes due to the chosen tolerances. This conclusion holds for all twelve trials, indicating that the operator used the same motion strategy in all trials: a pure alignment of peg and hole achieved by a planar motion, without exciting the two other relative degrees of freedom, being relative rotations about peg or hole axes. This is confirmed by Figure 2.14a where all the instantaneous screw axes and frames coincide.

Force invariants with respect to world: Figure 2.13b shows the screw invariants for the individual demonstration trials. Object invariant a, which corresponds to the magnitude of the force vector, is remarkably small, going from less than 4N to zero during the alignment. As for the moving frame invariants, only ω_κ is significantly different from zero, at least for six of the twelve trials. This indicates that, for those six trials the instantaneous screw

axis (i.e. the force vector) is rotating with respect to the world in a plane about a fixed point; for the other six trials the instantaneous screw axis remains fixed to the world, i.e. the force vector does not rotate. This in turn indicates that the operator used different wrench strategies in the trials.

Force invariants with respect to world and tool for averaged wrench trajectories: Based on the previous result, we hypothesized that the operator had used different wrench strategies in the two batches of demonstration trials, because they were executed at different points in time. Figure 2.13c therefore shows the screw invariants for the averaged wrench trajectories of batch 1 (red), batch 2 (blue), and the whole set of trials (green). We show both the invariants with respect to the world (solid lines) and to the tool (dashed lines). Only the first two rotational invariants are shown, as the other invariants are zero (v_b, v_t) or almost zero (ω_τ, b), similarly as in Figure 2.13b. Visually, there is an evident distinction between the trials of batch 1 and batch 2: while for batch 1 the average of the trials shows a force vector that rotates with respect to the world and remains fixed with respect to the peg, the opposite is true for the average of the trials of batch 2. To illustrate this further, we plot the resulting moving frames for one trial from batch 1 and one trial from batch 2 in Figures 2.14b and 2.14c. These figures clearly show that the ISA is rotating for the trial from batch 1 while the ISA is stationary for the trial from batch 2. A more in-depth study and statistical analysis of these wrench strategies go beyond the scope of this chapter, but we conclude that different strategies can lead to successful alignments. This is a valuable result for designing robot force control strategies. It is also interesting to see that it could be obtained in an experiment with only small contact forces.

2.7 Discussion and Conclusion

The aim of this work was to introduce local invariant descriptors for force and moment trajectories. Inspired by the invariant descriptors for motion, we exploited the duality between motion and force. We extended existing concepts for motion trajectories to general vector and screw trajectories, and then applied them to force and moment trajectories.

Two types of descriptors were introduced: vector and screw invariants. Vector invariants describe 3D vector trajectories (translational velocities, rotational velocities, forces, and moments) while screw invariants describe 6D screw trajectories (twists and wrenches). Both types of descriptors are invariant for the choice of world reference frame, but the screw invariants are also independent of the reference point on the moving body that is chosen to

represent the translational velocity or the moment. Hence, for obtaining the screw invariants there is no need to choose a reference point on the moving body. This is particularly useful in cases where there is no natural choice for such reference point, the reference point can change between executions, or the calibration of the reference point is time consuming. The newly proposed descriptors can be applied to both motion and force trajectories, as opposed to existing invariant descriptors, which were mainly intended for point curves and/or motion trajectories [27, 83, 116, 90, 87, 48, 47, 34, 62, 105].

Invariant properties allow to recognize and model tasks that were demonstrated in different environments, without the need for accurate calibration. For trajectory reconstruction and generation however, calibration efforts cannot be avoided since for these applications we need initial values as pointed out in the related paragraphs in Sections 2.3 and 2.4. More in detail, we need initial values for the orientation $\widetilde{\boldsymbol{R}}_1$ or location $\widetilde{\boldsymbol{T}}_1$ of the moving frame, and, if we reconstruct the object trajectory up to position and orientation, we also need respective initial values $\boldsymbol{p}_1$ and $\boldsymbol{R}_1$. In practical applications, these initial values are to be derived from environment models or from sensor measurements, either directly or using model-based state estimators.

Following a similar line of reasoning, the invariants themselves cannot be used to describe or check relations *between* motion and force, such as reciprocity between a screw twist and a screw wrench, unless the transformation between the respective moving frames in which the motion and force invariants are represented is known. In general, this requires calibration. For example, to check the *reciprocity condition*, i.e., the condition that no power is dissipated in the contact: $\boldsymbol{f} \cdot \boldsymbol{v} + \boldsymbol{m} \cdot \boldsymbol{\omega} = 0$, it is required that both twist and wrench, $\boldsymbol{t}$ and $\boldsymbol{w}$, are expressed in the same reference frame and have the same reference point. Therefore, we cannot just plug in the $\boldsymbol{a}$ and $\boldsymbol{b}$ object invariants of the respective screw invariants, because they are defined relative to their respective moving frames, and, in turn, to the frames in which the twist and wrench trajectories were measured. Hence, to study relations between motion and force, we need a calibration between the respective measurement frames.

The vector and screw invariants can be made independent of time, and hence independent of the applied time-based motion profile, by expressing them as a function of a suitable geometric progress variable. Again, this additional invariance is useful for the recognition of a task demonstrated in different situations and for the generalization of task models to new situations, as shown in [107] for the case of motion in free space. However, the way we chose the progress variable in the 3D contour following application introduced an important limitation. We did not follow the approach to choose an invariant progress variable that is independent of the reference point for translation, as suggested in [34] and briefly discussed in Section 2.2.3. Instead, we chose the

progress variable as the integral of the translational velocity of the origin of $\{tcp\}$. So, based on our understanding of the contact geometry, we handpicked this point as a reference point to define arc length as a representation of task progress, and used it to represent all invariants. The benefit of this approach is that we obtained vector and screw invariants for both motion and force that are interpretable and can be compared with each other and with the reference values of the limit case (Figures 2.6-2.8). Definitely, this progress value is a very good choice and probably even the best one for this application, but it would be better to derive a suitable progress value from the measured trajectories in an invariant data-driven way. This is subject of our future work.

Besides analytical procedures to derive the invariant descriptors, an approach was presented based on optimal control, where the main idea is to find the invariant descriptor that reconstructs the given window of measurements. This provides a way to deal with singularities and measurement noise and hence enables a much more robust calculation of the descriptors. Compared to our previous work on the calculation of motion descriptors [106], we improved the stability and repeatability of the calculation by reformulating the OCP. In the reformulated problem, trajectory reconstruction errors became inequality constraints that were bounded by a desired tolerance value. This resulted in more intuitive tuning, as we showed that these tolerance values could be directly related to the accuracy or noise characteristics of the used motion and force sensor. The reformulated OCP also resulted in a more stable behavior of the moving frame. For example, when the values of the noisy moments are below the specified tolerance, they can be described by a stationary moving frame (see Figure 2.7d and Figure 2.10b). Finally, the reformulation resulted in a smaller number of tuning parameters, further reducing the tuning effort. Overall, we obtained a good repeatability of the invariants, as can be seen in Figures 2.6-2.8, even for noisy data.

Continuity of the moving frames is ensured by minimizing the values of the moving frame invariants in the OCPs. This also provides some degree of continuity for the reconstructed trajectories. To enforce a more specific continuity on the reconstructed trajectory, it is advised to include the time profile as variables in the OCPs in order to enforce acceleration and/or jerk limit constraints. These constraints can be either on the spatial trajectory itself or on the robot joint trajectories in case a kinematic model of the robot is available, similar to [107].

The chapter contains a practical experiment involving a human-demonstrated 3D contour following task which showcases the use, verifies the invariant properties, and validates the robust calculation of screw and vector invariant descriptors for both motion and force. As expected, this experiment yielded highly repeatable motion descriptors due to the highly constrained motion (in

a 1-DOF instantaneous twist space), and less repeatable force descriptors due to human variation (in a 5-DOF instantaneous wrench space). Evidently, apart from human variation, other disturbances were present in this experiment such as sensor inaccuracy and measurement noise. A second experiment, involving human demonstration of a peg-on-hole alignment task, showed how motion and force invariants can be used to detect and characterize different human strategies in contact tasks.

The concepts and algorithms presented in this chapter are intended as an initial toolbox [7] to support future work on invariant representations of motion and force trajectories and their use in physics-informed machine learning of force-controlled manipulation skills.

Chapter 3

Automatic Derivation of an Optimal Task Frame for Learning and Controlling Contact-Rich Tasks*

Abstract

Choosing a proper reference frame for motion and wrench signals is crucial for successfully controlling contact-rich tasks. In previous work on learning of such tasks, this choice is often implicit, requires expert insight, or starts from predefined frame candidates. Furthermore, when an expert defines the frame,

*This chapter is based on: Mousavi Mohammadi, A. and Vochten, M., Aertbeliën, E., and De Schutter, J. Automatic derivation of an optimal task frame for learning and controlling contact-rich tasks, *arXiv preprint*, (2024). Ali Mousavi Mohammadi was the main and corresponding author for this paper. He contributed to the methodology development, formal analysis, investigation, experimental validation, data curation, software development, and drafting of the original manuscript. Maxim Vochten contributed to the methodology development, formal analysis, investigation, and the review and editing of the paper. Erwin Aertbeliën contributed to the methodology development, formal analysis, investigation, and the review and editing of the paper. Joris De Schutter provided supervision, contributed to the methodology development, formal analysis, investigation, and the review and editing of the paper. The terminology and notation have been adjusted slightly for consistency throughout the thesis. A supplementary video for this chapter is available at: `https://www.youtube.com/watch?v=I8-1FkOzOvU`

the exact location and/or orientation of the geometric features in the world or the tool that define this reference frame may be uncertain during the task demonstration or the task performance. This chapter presents an automatic method to derive the optimal reference frame, referred to as optimal task frame, directly from the recorded motion and wrench data of the demonstration. Using screw theory, several origin and orientation candidates are generated that maximize decoupling in the data with the purpose to design a decoupled controller. These candidates are then processed probabilistically, without needing hyperparameters, to obtain the optimal task frame. Its origin and orientation are independently fixed to either the world or the robot tool. The method works regardless of whether the task involves translation, rotation, force, or moment, or any combination thereof. The method was validated for a wide range of tasks, including surface following and manipulation of articulated objects, showing good agreement between derived and assumed expert task frames. To validate the robot's performance, a constraint-based controller was designed based on the data expressed in the derived task frames. These experiments demonstrated the approach's effectiveness and versatility. The automatic task frame derivation approach can support learning methods to design controllers for a wide range of contact-rich tasks.

3.1 Introduction

Contact-rich tasks in robotics are object manipulation tasks that involve contact with the environment or with other objects. In this chapter, we focus on the manipulation of a single object attached to the robot end effector in contact with another object attached to the environment. Such tasks are abundant in all fields of robotics: industry, domestic robots, health care, etc. They include, for example, assembly, machining, picking and placing, and handling of articulated objects such as doors or drawers.

Programming contact tasks on robot manipulators can be challenging since it requires coordinating and controlling both the desired motion of an object that is manipulated and its desired interaction wrench with the environment. Depending on the task, the translational and/or the rotational motion may be relevant, as well as the interaction force and/or the interaction moment.

Learning from Demonstration (LfD) can help to make the programming of contact tasks more intuitive since it allows non-expert users to teach tasks to robots without a deep knowledge of robotics [88]. Figure 3.1a illustrates a typical LfD workflow consisting of first recording the data of human demonstrations of the task, then learning a task model from the data and designing a corresponding

controller, and finally using this controller on the robot to perform the task while adapting to new circumstances.

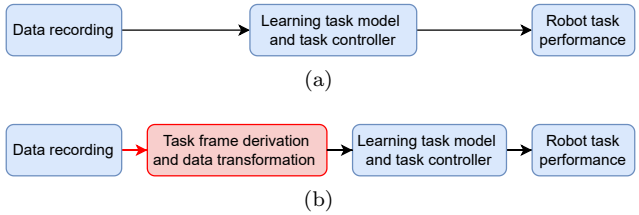


Figure 3.1: (a) The typical workflow in robot task learning consists of data recording, learning a task model and controller from the data, and performing the task on a robot. (b) Our approach adds the automatic derivation of an optimal task frame from the data, prior to learning the task model.

For object manipulation tasks in contact, it is important to define the task controller in a suitable reference frame. We refer to such a frame as a *task frame*. The task frame is defined as a coordinate frame, with an origin and an orientation, in which the motion and wrench signals are controlled in different directions towards desired values [23]. Given a suitable choice for the task frame, some of these signals become constant or zero, facilitating the design of a stable and robust controller. Therefore, the choice of the task frame is crucial for the success of the control strategy.

3.1.1 Problem and Research Objective

It is not always clear which choice of task frame is best for the controller. In some learning methods, the task frame is chosen implicitly to be the same as the reference frame in which the data were recorded. However, this usually does not lead to an optimal choice. Alternatively, the task frame can be chosen based on the expert user's intuition for the application at hand. For some tasks, the frame is best defined with respect to the robot's tool. Other times, it is better defined with respect to the environment. Considering the orientation, sometimes the axes of the frame are best chosen in the main direction of the translation or the interaction force, and other times they should be in the directions of rotation or interaction moment. Furthermore, the exact location of the geometric features in the world or the tool that define this reference frame according to the expert may be uncertain during the task demonstration or the task performance.

Some LfD approaches have aimed to assist the user by proposing several candidates for this reference frame, and using the demonstration data to decide

which candidate is most suitable [100, 57, 77, 70, 51]. However, this still requires defining such candidates beforehand.

A more user-friendly approach would be to derive a suitable task frame directly from the demonstration data itself. To the best of the authors' knowledge, this has not been explored yet. Most closely related are methods that learn the main directions of motion and force (or compliance) from the data in a given reference frame such as [97, 98, 96, 64, 63]. These main directions could be used to define the orientation of the task frame, as shown in [31] for force directions. However, since the directions are learned in a predefined reference frame, they are fixed to this reference frame, for example, the world or tool frame. Furthermore, deriving an optimal origin for the controller from data, i.e., the origin of the task frame, does not seem to have been considered yet in the literature.

The objective of this chapter is to propose a general method to automatically derive an optimal origin and orientation of the task frame from recorded motion and wrench data. To enable non-expert users to more easily learn a large variety of contact tasks, the method needs to have no restriction on the number of degrees of freedom in the task, and no prior assumption on whether the task is dominant in translation or rotation, or whether the task is dominant in force or moment.

3.1.2 Approach

Figure 3.1b highlights the step in the LfD workflow that we aim to optimize and automate. To derive the task frame, we first establish two principles for an optimal choice of the task frame based on maximizing the decoupling of signals in the controller. Based on these principles we generate multiple candidates for the origin and the orientation of the task frame from the demonstrated motion and wrench data, assuming that the data are segmented. We also assume that these candidates are either fixed to the world or to the tool. These candidates are processed to obtain the optimal task frame origin and orientation. Once the task frame has been derived, the desired motion and wrench signals are expressed in this frame and a controller can be designed to perform the task.

The automatic task frame derivation procedure is validated for a diverse set of contact-rich tasks, without any assumption on the type of task, by comparing the result with an expert's choice. The tasks are performed on the robot after designing a constraint-based controller based on the derived task frame. The controller's robustness and generalization capabilities are tested under disturbances, parameter variations, and alterations in the task frame.

3.1.3 Contribution and Outline

The main contribution of this research is a multi-step algorithm to derive an optimal task frame from demonstrated motion and wrench data. The main discerning features of our approach are that it incorporates data not only of the object's position and the interaction force, but also of the object's orientation and the interaction moment. It decides the most appropriate reference frame separately for the origin and orientation of the task. Finally, it decides the most appropriate variable in which the motion and force signals should be parameterized. The method is intended as a tool to automate the selection of a suitable reference frame before learning the task model with some learning method. The intent is to assist users that have limited experience in control.

The remainder of this chapter is organized as follows. Section 3.2 provides a more in-depth review of the literature on contact task learning related to the work proposed in this chapter. Section 3.3 presents mathematical preliminaries, while Section 3.4 proposes the methodology to automatically derive the optimal task frame from motion and wrench data. Sections 3.5 and 3.6 provide an extensive experimental validation of the task frame derivation procedure itself and of the use of the obtained task frame in the performance of learned tasks. Finally, Section 3.7 outlines the major contributions, limitations, and conclusions of the work.

3.2 Relation to the state of the art

3.2.1 Collection of Demonstration Data

Four common ways to collect demonstration data for LfD methods are kinesthetic teaching [64, 56, 119, 31, 97, 100, 51], teleoperation [79, 96], passive observation [80, 103] and simulation. The first method involves physically grasping the robot and moving it through the task, recording motion via the robot's encoders and interaction wrench via torque sensors at the joints or via a force/torque sensor at the wrist. While cost-effective and fast, this method can be challenging for heavy robots [96], bulky robots [41], redundant robots, dual-arm setups [94], or non-manipulator robots [88], as undesirable dynamics induced by the operator may affect the recorded data [80]. The second method is usually more expensive since it involves an external interface between the robot and the operator, such as a joystick, haptic device or graphical user interface. Collecting demonstration data by teleoperation is also more challenging than kinesthetic teaching [43]. In the third method, the operator uses an instrumented tool to perform the task and record motion and wrench data. In the last method,

simulation environments like MoJuCo [5] and Bullet [3] are used to generate demonstration data. The latter method offers abundant data which can enable reinforcement-learning-based methods to widely explore the task without any risk for the robot. However, transferring learned skills from the simulator to the real world is still challenging [95]. Consequently, building a reliable model derived from one [64, 56, 63] or a few [103, 119, 100] demonstrations is a critical aspect of learning contact-rich tasks. In this chapter, passive observation is employed to record demonstrations as it aligns with the most natural way for the operator to perform tasks [41], but the developed approach is not limited to this demonstration method. Furthermore, the proposed approach to derive an optimal task frame can also be used with simulation data as input or with robot data, i.e., motion and wrench data recorded during a task performed by a robot.

3.2.2 Assumptions and Applications

Some researchers included methods for segmentation in their proposed pipelines for processing the demonstration data, such as [100, 57, 64, 63], but we left segmentation out of the scope of the present chapter. I.e., we assume demonstration data has already been segmented and propose an approach to derive a task frame for a single task segment in which the contact situation remains the same over time. On the other hand, while a lot of approaches in literature focus on LfD for a single task or a limited class of tasks, our aim was to develop a generic approach, without further knowledge or assumptions about the task other than that the contact is between only two objects with a measurable relative motion and interaction wrench.

We further assume that the demonstration is provided by a skilled operator who performs the task efficiently. Specifically, we expect that the operator (1) excites only the degrees of freedom necessary to accomplish the task and (2) applies a reasonable amount of force and torque, sufficient to maintain contact and achieve the goal without unnecessary effort. For instance, in a Peg on Hole Alignment task[2], a skilled operator would not rotate the peg about peg's symmetry axis unnecessarily while aligning it with the hole. At the same time, the operator would apply sufficient force to ensure continuous contact without exerting excessive force, i.e. force that is not needed to successfully complete the task. These are natural assumptions, as humans generally avoid complicating tasks unnecessarily or exerting significant amount of force or torque while it is not required.

[2]This task is detailed in Section 3.5.1.

3.2.3 Selection of Task Frame

To control a robot while manipulating an object, typically, a proper reference frame is chosen. This reference frame has different names in the literature such as *compliance frame* [71], *constraint frame* [31], and *task frame* [23] but the main idea behind it remains the same. It was first proposed by Mason [71], where a task was specified in terms of natural (i.e. physical) and artificial (i.e. imposed) constraints for motion and wrench in the context of frictionless contact between rigid bodies. Based on this idea, Raibert and Craig [86] formulated a hybrid controller to obtain decoupling between the position- and force-controlled directions. The task frame is selected to simplify the specification of the natural and artificial constraints. Since this process requires a good understanding of the task geometry, it is entrusted to the expert [86]. De Schutter and Van Brussel [37, 38] introduced an adaptive or *moving* task frame for task geometries such as in 2-D contour following, where the directions of contact force and translational velocity are not fixed with respect to both world and tool, but are estimated from the motion and wrench measurements during the task performance. Bruyninckx et al. [24] presented a generic task geometry involving contact between two objects with doubly curved surfaces and leading to a contact frame that moves with respect to *both* objects in contact. This moving frame simplifies for special cases such as contact between polyhedral objects. Bruyninckx and De Schutter provided more examples of task frame selections, including adaptive and non-adaptive ones [23]. They suggested to maximize the decoupling between motion and wrench directions, but did not present a detailed method for achieving this goal.

Task frame selection regained interest in the context of LfD, since demonstration data and task models have to be represented in an appropriate reference frame. There are two common ways to choose a task frame. The first way is, still as in earlier works, to rely on implicit high-level robotics expertise, i.e., without using any prescribed method. The second way is to select between several task frame candidates that are predefined based on expert knowledge. This selection is based on criteria like consistency [100], variance [57] or generalization capability of the demonstration data [70]. A first set of such studies *directly* select between different task frame candidates [100, 57, 77] or optimize the position and orientation of such candidates [51]. Other studies *indirectly* learn an appropriate reference frame or components thereof. For example, a method was proposed which learns to activate a set of attractors that are each defined in a different appropriate reference frame [70]. In [31], the authors proposed a method that could adapt the orientation of the task frame with respect to the environment. Additionally, some works learned the main directions of motion and force (or compliance) from the data in a given reference frame, such as [97, 98, 96, 64, 63].

In this chapter, we propose a *generic* approach to derive an optimal task frame for contact-rich tasks, allowing interaction with both articulated and non-articulated objects. Both the origin and the orientation of the task frame are derived. The origin and orientation can be fixed in different viewpoints (world or tool) so that the task frame as a whole is not necessarily fixed to either world or tool. We use both motion and wrench data, consider both their translational and rotational vector components and automatically derive the motion and wrench *vector components of interest* from the demonstration data. The procedure does not involve hyperparameters. While all works mentioned above exhibit some of these features, they lack several others. In particular, to the best of the authors' knowledge, there is no method available in the literature to derive the task frame from motion and wrench data. Others use application-specific knowledge [39, 97, 98, 80, 79, 8, 96, 84, 31, 103, 93, 119, 64, 63] or predefined candidates [77, 57, 100, 51, 70]. Some works do not consider tasks involving articulated objects [119, 103, 31, 84, 96, 70, 51, 80, 98, 97, 100, 57, 77, 93] or non-articulated objects [63, 64]. Some works do not consider tasks with dominant rotation [100, 97, 98, 80, 79, 8, 70, 96, 84, 31, 119, 64, 63], force [77, 51, 70] or moment [77, 100, 85, 97, 98, 80, 79, 70, 96, 8, 51, 84, 31, 119, 64, 63]. On the other hand, although it is not common to consider a *moving* task frame for contact-rich tasks in the context of LfD, this was done in [31].

In this chapter, our objective is to identify a *single* task frame for each task segment. Representing the data within such a unified frame simplifies both task modeling and controller design, as it enables consistent interpretation and management of motion and wrench information. The use of a single task frame is common in the literature and has proven effective in practice. In contrast, employing multiple task frames increases complexity and makes it more challenging to resolve potential conflicts between motion and force data. We explored the use of two separate task frames but quickly found that it became difficult to perform the task effectively. This experience led us to combine the frames and perform the task using a single, unified task frame. This frame therefore provides a coherent basis for analysis and facilitates the development of robust control strategies.

3.2.4 Task Model and Control

Typically, for learning contact-rich tasks from human demonstration, motion and wrench data are recorded to build a task model. However, in [80] muscle activity (EMG signals) was recorded as well to estimate the desired stiffness behavior of the robot end-effector. Various methods have been explored to model skills required for contact-rich tasks. These models should be able to reproduce the task under similar and novel situations. Some of the most common

methods are Gaussian mixture model/regression [8, 119] and dynamic movement primitives [80, 84, 31, 39]. From the learned models the desired values for motion and interaction wrench are generated. These signals need to be tracked by a controller. Unlike position/force control, as applied in [31], impedance control is a popular solution for contact-rich tasks, with different types of controllers employed in [100, 39, 85, 98, 80, 8, 84, 119]. However, impedance behaviors during contact can also be achieved using an indirect formulation, as shown in [103]. Finally, the presence of hard geometric constraints is searched for in the demonstration data in [64, 63], while such constraints are explicitly defined in the task model in [79].

The task model used in the experimental validation Sections 3.5 and 3.6 is derived from 6-D motion and 6-D wrench data. It consists of 1) a task frame, with its location and orientation as parameters with fixed values, and 2) the demonstration data, expressed in the task frame and averaged over the demonstration trials to find the desired values for the motion and interaction wrench. No further processing of the data is performed. A *generic* constraint-based controller is used to follow these desired values. However, we believe that other learning methods could leverage this automatically derived task frame, for building their models and exploring alternative controllers.

3.2.5 Generalization and Skill Transfer

The learned model is expected to be able to perform the task not only under conditions similar to the demonstrations but also in novel scenarios. This challenge is commonly referred to as generalization. Recent studies explored aspects such as starting from new positions [100, 57, 97, 84], reaching to new positions [100, 57], positional displacements [85, 64], dealing with unexpected end-effector collisions [119, 31], variations in contact friction [100, 31], and achieving faster performance of the learned task [103]. Transferring learned skills to new objects and geometries is another crucial aspect. The model should be able to perform the task with similar but different objects, as discussed in [100, 57, 79, 80, 103, 64, 55, 56]. This involves preserving the key features of the demonstrated task while achieving the desired task goals. In this chapter, we also show how the proposed approach can be used to generalize acquired skills to novel scenarios and to transfer these skills to new objects using the parameters of the learned task model.

3.3 Preliminaries and Notation

3.3.1 Orientation and Pose

In this chapter we distinguish between two types of reference frames that can be attached to a rigid body $\mathcal{B}$. The first type is referred to as an *orientation frame* since only the orientation of this frame is relevant, not the origin. A general orientation frame $[b]$ can be represented with respect to another frame $[a]$ by a rotation matrix ${}^{b}\boldsymbol{R}_{a} \in \mathbb{R}^{3\times3}$:

$$ {}^{b}\boldsymbol{R}_{a} = \begin{bmatrix} {}^{b}\boldsymbol{e}_{x\,a} & {}^{b}\boldsymbol{e}_{y\,a} & {}^{b}\boldsymbol{e}_{z\,a} \end{bmatrix}, \tag{3.1} $$

where the columns ${}^{b}\boldsymbol{e}_{x\,a}$, ${}^{b}\boldsymbol{e}_{y\,a}$, and ${}^{b}\boldsymbol{e}_{z\,a}$ are the coordinates of the three orthogonal basis vectors of $[b]$ expressed in $[a]$.

Equation (3.1) represents the orientation of $[b]$ with respect to $[a]$ using coordinates expressed in $[a]$. To analyze the same relative rotation, but using coordinates expressed in a third frame $[c]$, a *similarity transformation* can be used [92]:

$$ {}^{b}_{c}\boldsymbol{R}_{a} = {}^{c}\boldsymbol{R}_{a}^{-1}\ {}^{b}\boldsymbol{R}_{a}\ {}^{c}\boldsymbol{R}_{a}. \tag{3.2} $$

The three successive rotation matrices are interpreted as follows: change the reference frame from $[c]$ to $[a]$ using ${}^{c}\boldsymbol{R}_{a}^{-1}$, then apply the relative rotation ${}^{b}\boldsymbol{R}_{a}$ in frame $[a]$, and finally change the reference frame back to $[c]$ using ${}^{c}\boldsymbol{R}_{a}$. In robotics such a similarity transformation is also used to transform the coordinate system of an inertia matrix or screw matrix, which is better known as an *adjoint transformation* or an *adjoint map* [76, 68].

The orientation of a frame can also be represented by a *rotation vector* $\boldsymbol{r} \in \mathbb{R}^{3\times1}$. This rotation vector has the meaning of a rotation about an axis along the unit vector $\boldsymbol{r}/\left\|\boldsymbol{r}\right\|$ by an angle $\left\|\boldsymbol{r}\right\|$. The rotation matrix and rotation vector are related through the matrix exponential:

$$ \boldsymbol{R} = \exp\left([\boldsymbol{r}]_{\times}\right), \tag{3.3} $$

where $[\cdot]_{\times}$ is an operator that produces the skew-symmetric cross product matrix from a vector. The inverse relation is established by the matrix logarithm:

$$ \left(\log \boldsymbol{R}\right)^{\vee} = \boldsymbol{r}, \tag{3.4} $$

where the operator $(\cdot)^{\vee}$ is the inverse of $[\cdot]_{\times}$ such that $\boldsymbol{r} = ([\boldsymbol{r}]_{\times})^{\vee}$.

The second type of reference frame is referred to as a *pose frame*. Both its orientation and its origin are relevant. The orientation of a pose frame $\{b\}$ can again be represented by the rotation matrix ${}^{b}\boldsymbol{R}_{a}$, while the position of its origin

with respect to $\{a\}$ is given by a position vector ${}^{b}\boldsymbol{p}_{a} \in \mathbb{R}^{3\times 1}$. The orientation and position of the pose frame can be combined in a 4×4 *homogeneous transformation matrix* or *pose matrix*:

$$ {}^{b}\boldsymbol{T}_{a} = \begin{bmatrix} {}^{b}\boldsymbol{R}_{a} & {}^{b}\boldsymbol{p}_{a} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix}. \tag{3.5} $$

To analyze the relative pose between $\{b\}$ and $\{a\}$ using coordinates expressed in a third frame $\{c\}$, we extend the *similarity transformation* of (3.2) to pose matrices:

$$ {}^{b}_{c}\boldsymbol{T}_{a} = {}^{c}\boldsymbol{T}_{a}^{-1}\ {}^{b}\boldsymbol{T}_{a}\ {}^{c}\boldsymbol{T}_{a}. \tag{3.6} $$

Figure 3.2 visualizes the similarity transformation under general conditions with three moving frames.

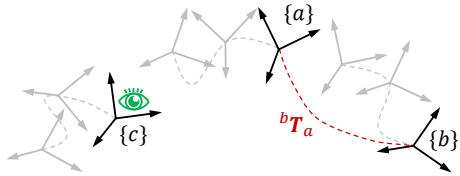


Figure 3.2: The relative pose ${}^{b}\boldsymbol{T}_{a}$ between frames $\{a\}$ and $\{b\}$ can be observed from the perspective of a third frame $\{c\}$. Without loss of generality, any of these frames could be stationary.

The pose of a frame can also be represented by a *displacement screw* $\boldsymbol{\mathcal{d}} = (\boldsymbol{r}^{T}\ \ \boldsymbol{u}^{T})^{T} \in \mathbb{R}^{6\times 1}$, where $\boldsymbol{r}$ is again the rotation vector and $\boldsymbol{u}$ is the translation at the origin of the reference frame. They are related by the matrix exponential:

$$ \boldsymbol{T} = \exp\left([\boldsymbol{\mathcal{d}}]_{\times}\right), \tag{3.7} $$

where $[\cdot]_{\times}$ is an operator that produces the following matrix from a screw:

$$ [\boldsymbol{\mathcal{d}}]_{\times} = \begin{bmatrix} [\boldsymbol{r}]_{\times} & \boldsymbol{u} \\ \boldsymbol{0}_{1\times 3} & 0 \end{bmatrix}. \tag{3.8} $$

The inverse relation is established by the matrix logarithm:

$$ (\log \boldsymbol{T})^{\vee} = \boldsymbol{\mathcal{d}}, \tag{3.9} $$

where the operator $(\cdot)^{\vee}$ applied to a screw is defined as the inverse of $[\cdot]_{\times}$ such that $\boldsymbol{\mathcal{d}} = ([\boldsymbol{\mathcal{d}}]_{\times})^{\vee}$.

3.3.2 Velocity Kinematics

The velocity kinematics of an orientation frame are characterized by the rotational velocity vector $\boldsymbol{\omega}$. When the vector's coordinates are expressed in $[a]$, the relation between ${}_a\boldsymbol{\omega}$ and the derivative of the rotation matrix ${}^b\dot{\boldsymbol{R}}_a$ is:

$$ {}^b\dot{\boldsymbol{R}}_a = [{}_a\boldsymbol{\omega}]_\times \, {}^b\boldsymbol{R}_a. \tag{3.10} $$

The velocity kinematics of a pose frame are characterized by the *twist* $\boldsymbol{t} = (\boldsymbol{\omega}^T \quad \boldsymbol{v}^T)^T \in \mathbb{R}^{6\times 1}$, consisting of the rotational velocity vector $\boldsymbol{\omega}$ and the translational velocity vector $\boldsymbol{v}$ at the origin o of the reference frame. When the twist coordinates are expressed in $\{a\}$, the relation between ${}_a\boldsymbol{t}$ and the time-derivative of the pose matrix ${}^b\dot{\boldsymbol{T}}_a$ can be worked out as:

$$ {}^b\dot{\boldsymbol{T}}_a = [{}_a\boldsymbol{t}]_\times \, {}^b\boldsymbol{T}_a. \tag{3.11} $$

3.3.3 Wrench and General Screw

The interaction force and moment between two rigid bodies can be represented using a *wrench* $\boldsymbol{w} = (\boldsymbol{f}^T \quad \boldsymbol{m}^T)^T \in \mathbb{R}^{6\times 1}$, where $\boldsymbol{f}$ and $\boldsymbol{m}$ are the interaction force vector and interaction moment vector at the origin o of the reference frame respectively.

The twist and wrench can both be interpreted as *screws* according to screw theory [15]. For twists, the theorem of Mozzi-Chasles holds that the motion of a rigid body can always be represented as a rotation and translation along an axis in space, the *screw axis for motion* [76, 42]. Similarly, Poinsot's theorem states that all forces and moments acting on a rigid body can always be represented as a force and moment along an axis in space, the *screw axis for wrench*.

Both the screw twist and wrench will be represented in this chapter with a *general screw* $\boldsymbol{\mathcal{s}} = (\boldsymbol{a}^T \quad \boldsymbol{b}^T)^T$ where $\boldsymbol{a}$ is the directional component of the screw along the screw axis, $\boldsymbol{b}$ is the moment component of the screw, i.e. the moment about the screw axis seen at the origin of the reference frame, and the coordinates of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are expressed in the reference frame.

A screw $\boldsymbol{\mathcal{s}}$ that is expressed in reference frame $\{b\}$ can be transformed to $\{a\}$ using a *screw transformation matrix* $\boldsymbol{S}$:

$$ {}_a\boldsymbol{\mathcal{s}} = \boldsymbol{S}\left({}^b\boldsymbol{T}_a\right) \, {}_b\boldsymbol{\mathcal{s}}, \tag{3.12} $$

with:

$$\boldsymbol{S}\left({}^{b}\boldsymbol{T}_{a}\right)=\begin{bmatrix}{}^{b}\boldsymbol{R}_{a} & \boldsymbol{0}_{3\times3}\\ \left[{}^{b}\boldsymbol{p}_{a}\right]_{\times}{}^{b}\boldsymbol{R}_{a} & {}^{b}\boldsymbol{R}_{a}\end{bmatrix}. \tag{3.13}$$

Note that applying this transformation for screws bears a relation with the similarity transformation for poses (3.6), as also explained in [76, 68].

A special case of (3.12) where only the application point of the moment component $\boldsymbol{b}$ needs to be changed from a point ${}^{o_1}\boldsymbol{p}$ to ${}^{o_2}\boldsymbol{p}$, can succinctly be written as:

$$\boldsymbol{b}^{o_2}=\left[{}^{o_2}\boldsymbol{p}-{}^{o_1}\boldsymbol{p}\right]_{\times}\boldsymbol{a}+\boldsymbol{b}^{o_1}, \tag{3.14}$$

in which the coordinates of all vectors are expressed with respect to the same reference frame.

3.3.4 Time-invariant Motion and Wrench Trajectories

Motion and wrench trajectories are usually considered as a function of time t, such as the pose $\boldsymbol{T}(t)$, twist $\boldsymbol{t}(t)$, and wrench $\boldsymbol{w}(t)$ trajectories. They can be made time-invariant by re-parameterizing them as a function of a geometric progress variable ξ. There are multiple valid choices for the progress variable. For example, it can be defined as the cumulative arc length of the trajectory traversed by a chosen reference point, or as the cumulative rotation angle, or by some combination of the two. The progress rate $\dot{\xi}$ would then correspond to the traversed arc length or rotation angle per unit time, or their combination. Often, the progress variable is also made non-dimensional (ranging from 0 to 1) so that trajectories of different length can be compared.

For a given progress $\xi(t)$ as a function of time, the reparameterizations of pose, wrench, and twist from the time domain to the progress domain are defined as follows [108]:

$$\boldsymbol{T}(\xi)=\boldsymbol{T}(t(\xi)),\ \boldsymbol{w}(\xi)=\boldsymbol{w}(t(\xi)),\ \boldsymbol{t}(\xi)=\frac{\boldsymbol{t}(t(\xi))}{\dot{\xi}(t(\xi))}, \tag{3.15}$$

where $t(\xi)$ is the inverse function of $\xi(t)$.

We can go back from the progress domain to the time domain using:

$$\boldsymbol{T}(t)=\boldsymbol{T}(\xi(t)),\ \boldsymbol{w}(t)=\boldsymbol{w}(\xi(t)),\ \boldsymbol{t}(t)=\boldsymbol{t}(\xi(t))\dot{\xi}(t). \tag{3.16}$$

Consecutive application of (3.15) and (3.16) with different progress timings $\xi(t)$ results in a different time trajectory $\boldsymbol{T}(t)$, i.e., the time-dependency of the trajectory has changed, but the spatial path has remained the same.

These re-parameterizations are helpful to master the variability in demonstrations by allowing us to make time-invariant models. In practice, for discrete-time data, these reparameterizations are performed using numerical interpolation. Throughout this chapter and in order not to overload notation, we do not explicitly mention the dependence of signals on time or on another progress variable when such dependence is obvious from the context.

3.3.5 Average Vector Orientation Frame (AVOF)

The *Average Vector Orientation Frame* (AVOF) is an orientation frame that characterizes the average direction and main directions of variation for a given set of N unordered vectors $\{\boldsymbol{c}_i\}$, with $i = 1 \dots N$. The concept was originally introduced in [10] for characterizing the (variation in) orientation of the screw axis corresponding to a set of twists $\{\boldsymbol{t}_i\}$. Later it was extended in [108] to characterize sets of general vectors $\{\boldsymbol{c}_i\}$.

The AVOF is calculated as follows. First construct the *uncentered* covariance matrix from the vectors $\{\boldsymbol{c}_i\}$:

$$\boldsymbol{C}_c = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{c}_i \boldsymbol{c}_i^T. \tag{3.17}$$

(The data is uncentered so that the average direction of the vectors is included.) Next, the *Singular Value Decomposition* (SVD) of the covariance matrix $\boldsymbol{C}_c$ is determined. Since $\boldsymbol{C}_c$ is a symmetric matrix, the SVD can be simplified to:

$$\boldsymbol{C}_c = \boldsymbol{U}_{avof} \; \boldsymbol{\Sigma}_{avof} \; \boldsymbol{U}_{avof}^T, \tag{3.18}$$

where $\boldsymbol{U}_{avof}$ is a right-handed orthonormal matrix consisting of three singular vectors and $\boldsymbol{\Sigma}_{avof}$ is a diagonal matrix containing the singular values of $\boldsymbol{C}_c$: $\sigma_1^2 \geq \sigma_2^2 \geq \sigma_3^2$.

The AVOF's orientation corresponds to $\boldsymbol{U}_{avof}$. The first column of $\boldsymbol{U}_{avof}$ is interpreted as the average direction of all vectors $\{\boldsymbol{c}_i\}$. The second and third columns are the main directions of variation for the vectors. The overall magnitude of the vectors in these three directions is given by σ_1, σ_2, and σ_3, respectively. Figure 3.3 provides an illustration of the AVOF where the spatial distribution of vectors is visualized using a dispersion ellipsoid corresponding to the SVD.

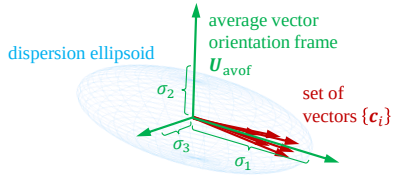


Figure 3.3: Average vector orientation frame for a set of vectors $\{\boldsymbol{c}_i\}$ together with a dispersion ellipsoid characterizing the main directions of variation.

The difference between the AVOF method and other SVD-based approaches in the literature [100, 97, 31, 64] is that in the AVOF method there is no explicit normalization of the vectors $\boldsymbol{c}_i$ before calculating the covariance matrix $\boldsymbol{C}_c$. The reason is that additive noise on a vector with a small magnitude significantly alters the direction of the normalized vector. To remove this sensitivity, the vectors $\boldsymbol{c}_i$ are not normalized, so that in the SVD more weight is given to vectors with higher magnitudes than vectors with lower magnitudes. This can be interpreted as a weighted average of directions, where the weights correspond to the magnitudes.

The variation on the estimated orientation of the AVOF, in terms of a covariance matrix $\boldsymbol{C}_{avof}$, can be defined by scaling the covariance matrix $\boldsymbol{C}_c$ so that it becomes dimensionless. There are different options for scaling. Here, we divide $\boldsymbol{C}_c$ by the trace of $\boldsymbol{C}_c$, which is the same as the trace of $\boldsymbol{\Sigma}_{avof}$ and also the same as the average squared norm of the vectors $\boldsymbol{c}_i$:

$$\boldsymbol{C}_{avof} := \frac{\boldsymbol{C}_c}{\mathrm{trace}(\boldsymbol{C}_c)} = \frac{\boldsymbol{C}_c}{\mathrm{trace}(\boldsymbol{\Sigma}_{avof})} = \frac{\boldsymbol{C}_c}{\frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{c}_i\|^2}. \tag{3.19}$$

The calculation of the AVOF orientation $\boldsymbol{U}_{avof}$ and the covariance matrix $\boldsymbol{C}_{avof}$ will be referred to by the function operator `avof` in the rest of this chapter. Algorithm 1 summarizes the AVOF method.

Algorithm 1: Average Vector Orientation Frame

Function `avof(` $\{\boldsymbol{c}_i\}$`):`
  Input: $\{\boldsymbol{c}_i\}$ with $i = 1 \dots N$; `// set of vectors`
  Output: $\boldsymbol{U}_{avof}$, $\boldsymbol{C}_{avof}$;
  $\boldsymbol{C}_c \leftarrow \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{c}_i \boldsymbol{c}_i^T$; `// covariance matrix`
  $\boldsymbol{U}_{avof}, \boldsymbol{\Sigma}_{avof}, \ldots \leftarrow \mathrm{svd}(\boldsymbol{C}_c)$;
  $\boldsymbol{U}_{avof}(:,3) \leftarrow \boldsymbol{U}_{avof}(:,1) \times \boldsymbol{U}_{avof}(:,2)$; `// right-handed`
  $\boldsymbol{C}_{avof} \leftarrow \boldsymbol{C}_c / \mathrm{trace}(\boldsymbol{C}_c)$; `// make nondimensional`

3.3.6 Average Screw-axes Intersection Point (ASIP)

The *Average Screw-axes Intersection Point* (ASIP) is the approximate intersection point for a given set of N unordered screw axes associated with screws $\{\boldsymbol{\$}_i\}$ with $i = 1 \dots N$. In [10], a method was proposed to calculate the ASIP for a set of screw twists $\{\boldsymbol{t}_i\}$ by finding the point with minimal average velocity due to all screws. By considering the velocity instead of the distance to the screw axes, the resulting estimate of the ASIP was shown to be more robust to noise. This is because, in the presence of additive noise, the screw axis is not well-defined when the directional component $\boldsymbol{a}_i$ is very small. Later, the method was extended to general screws $\{\boldsymbol{\$}_i\}$ in [108], so that it could be applied to wrenches $\{\boldsymbol{w}_i\}$ in addition to twists $\{\boldsymbol{t}_i\}$.

The ASIP for a set of screws $\{\boldsymbol{\$}_i\}$ is defined as the point with *minimal average moment* (i.e., translational velocity for $\{\boldsymbol{t}_i\}$ or moment for $\{\boldsymbol{w}_i\}$). This point can be calculated by solving the following least-squares problem:

$$\boldsymbol{p}_{asip} = \underset{\boldsymbol{p}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left\| \boldsymbol{a}_i \times \boldsymbol{p} + \boldsymbol{b}_i \right\|^2 . \tag{3.20}$$

This problem is ill-defined when all the directional vectors $\{\boldsymbol{a}_i\}$ of the screws are parallel or close to zero. Therefore, a regularization term $\epsilon \left\| \boldsymbol{p} - \boldsymbol{p}_0 \right\|^2$ is added in the above criterion (3.20) with a weight ϵ that biases the solution towards a given prior value $\boldsymbol{p}_0$. When ϵ is kept small, the effect is negligible in the regular case. The solution of the regularized least-squares problem can be worked out as:

$$\boldsymbol{p}_{asip} = \left[\boldsymbol{A} + \epsilon \boldsymbol{I}_3 \right]^{-1} \left(\frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{a}_i \times \boldsymbol{b}_i) + \epsilon \, \boldsymbol{p}_0 \right), \tag{3.21}$$

with $\boldsymbol{A} = \frac{1}{N} \sum_{i=1}^{N} [\boldsymbol{a}_i]_\times [\boldsymbol{a}_i]_\times^T$ and $\boldsymbol{I}_3$ a 3×3 identity matrix.

The variation on the estimate of the ASIP can be studied using the following covariance matrix [10]:

$$\boldsymbol{C}_{asip} = \operatorname{var}(\boldsymbol{p}_{asip}) = \hat{\sigma}^2 \left(\boldsymbol{A} + \epsilon \boldsymbol{I}_3 \right)^{-1}, \tag{3.22}$$

where the value of $\hat{\sigma}^2$ can be estimated from the residuals on the moment in (3.20):

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{N} \left\| \boldsymbol{a}_i \times \boldsymbol{p}_{asip} + \boldsymbol{b}_i \right\|^2}{N(3N-3)}. \tag{3.23}$$

Similar to the AVOF, the main directions of variation can be extracted using an SVD on the covariance matrix $\boldsymbol{C}_{asip}$:

$$\boldsymbol{C}_{asip} = \boldsymbol{U}_{asip} \, \boldsymbol{\Sigma}_{asip} \, \boldsymbol{U}_{asip}^T, \tag{3.24}$$

with $\boldsymbol{U}_{asip}$ the matrix of singular vectors and $\boldsymbol{\Sigma}_{asip}$ a diagonal matrix containing the singular values $\boldsymbol{C}_{asip}$. The first column of $\boldsymbol{U}_{asip}$ is interpreted as the direction in which the estimation of the ASIP is the most uncertain. Figure 3.4 illustrates the ASIP for a set of screws where the corresponding directions of variation are illustrated by an uncertainty ellipsoid.

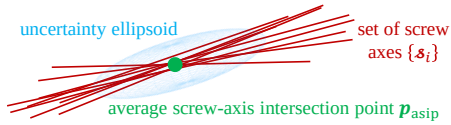


Figure 3.4: Average Screw-axes intersection point $\boldsymbol{p}_{asip}$ for a set of screw axes $\{\mathcal{S}_i\}$ and its corresponding uncertainty ellipsoid.

The calculation of the ASIP position, $\boldsymbol{p}_{asip}$, the singular values $\boldsymbol{\Sigma}_{asip}$, the covariance matrix $\boldsymbol{C}_{asip}$, and the matrix of singular vectors $\boldsymbol{U}_{asip}$ will be referred to by the function operator `asip` in the rest of this chapter. Algorithm 2 summarizes the ASIP method.

Algorithm 2: Average Screw-axes Intersection Point

Function `asip(` $\{\mathcal{S}_i\}$`):`
  Input: $\{\mathcal{S}_i\}$ with $i = 1 \dots N$; `// set of screw axes`
  Output: $\boldsymbol{p}_{asip}$, $\boldsymbol{C}_{asip}$;
  $\boldsymbol{A} \leftarrow \frac{1}{N}\sum_{i=1}^{N}[\boldsymbol{a}_i]_\times[\boldsymbol{a}_i]_\times^T$;
  $\boldsymbol{p}_{asip} \leftarrow (\boldsymbol{A}+\epsilon\boldsymbol{I}_3)^{-1}\left[\frac{1}{N}\sum_{i=1}^{N}([\boldsymbol{a}_i]_\times\boldsymbol{b}_i)+\epsilon\boldsymbol{p}_0\right]$;
  $\hat{\sigma}^2 \leftarrow \frac{\sum_{i=1}^{N}\|\boldsymbol{a}_i\times\boldsymbol{p}_{asip}+\boldsymbol{b}_i\|^2}{N(3N-3)}$;
  $\boldsymbol{C}_{asip} \leftarrow \hat{\sigma}^2(\boldsymbol{A}+\epsilon\boldsymbol{I}_3)^{-1}$; `// covariance matrix`

Relation between AVOF and ASIP: When the AVOF method is applied to the directional component $\{\boldsymbol{a}_i\}$ of a set of screws $\{\mathcal{S}_i\}$, the following relationship can be established between the covariance matrix of the AVOF and the covariance matrix of the ASIP [10]:

$$\boldsymbol{C}_{asip} = \frac{\hat{\sigma}^2}{\text{trace}(\boldsymbol{C}_c)}(\boldsymbol{I}_3 - \boldsymbol{C}_{avof})^{-1}. \tag{3.25}$$

Consequently, the AVOF and ASIP have the same directions of variation since they share the same set of singular vectors. Additionally, from this equation it follows that the highest degree of uncertainty for the ASIP will always be in the dominant direction of the screw axes, as also visualized in Figure 3.4.

3.4 Derivation of the Task Frame

This section outlines the proposed method to derive the task frame. We first delineate the input data and expected outcome. Then the global approach is explained, followed by the detailed procedures to select the origin and orientation of the task frame. Finally we provide details on how to represent the demonstration data in the obtained task frame.

3.4.1 Input Data and Expected Outcome

The inputs for deriving the task frame are the motion and wrench data collected during one or more demonstration trials. To ensure a representative task frame, care has to be taken that the demonstrations sufficiently excite the degrees-of-freedom (DOF) in motion and interaction wrench that are expected to be present in the actual performance of the learned task by the robot. Typically, the data is represented as a function of discrete time. We assume the data are segmented, i.e. they start at the beginning and finish at the end of the motion in contact to be modeled. Other assumptions are that the measured wrench corresponds to the interaction wrench, i.e. gravity is compensated and inertial forces are negligible, and the measured poses are sufficiently smooth such that smooth twists signals can be derived.

In general, the task frame $\{tf\}$ can be a freely moving frame. However, this chapter limits itself to finding the best possible task frame that has: (1) a fixed origin with respect to either a reference frame $\{w\}$ attached to the world $\mathcal{W}$ or with respect to a reference frame $\{tl\}$ attached to the tool $\mathcal{TL}$, and (2) a fixed orientation with respect to either $\{w\}$ or $\{tl\}$. We refer to this as selecting the *viewpoints*, $\mathcal{W}$ or $\mathcal{TL}$, for the origin and the orientation. Despite these limitations, such an approach is already suitable for a wide variety of practical applications, as abundantly shown in literature and demonstrated in Sections 3.5 and 3.6.

Consequently, the expected outcome of the task frame derivation procedure consists of: (1) the viewpoint for the origin ($\mathcal{W}$ or $\mathcal{TL}$) and the position vector representing the origin in the corresponding reference frame ($\{w\}$ or $\{tl\}$), i.e. ${}^{tf}\boldsymbol{p}_{*}$ where '$*$' is w or tl, and similarly, (2) the viewpoint for the orientation and the rotation matrix (or other representation) representing the orientation with respect to the corresponding reference frame, i.e. ${}^{tf}\boldsymbol{R}_{\circ}$ where '$\circ$' is w or tl.

With the task frame defined, the motion and wrench data can be represented in this frame to serve as a starting point for specifying desired values for the controller. To allow representation of the data in a time-independent

way, a suitable geometric progress variable ξ has to be chosen, as explained in Section 3.3.4. Typically, the progress rate $\dot{\xi}$ is chosen based on the instantaneous translational velocity, rotational velocity or a combination of both. This choice is made depending on the application and is based on an expert's insight. Therefore, as a third outcome of the procedure, we aim to determine an appropriate progress rate for the application in a data-driven way.

The whole procedure should deliver invariant outcomes, i.e. results that are independent of the chosen data representation, and involve as few hyperparameters as possible to limit the required expert knowledge to tune them. If hyperparameters are needed, they are preferably as much as possible application-independent, to limit the tuning effort across different applications. Also, the outcomes should be easily transferable from the demonstration setup to the robot setup while adequately dealing with the differences between both.

3.4.2 Conceptual Approach

Inspired by the related work reviewed in Section 3.2.3, we hypothesize that the underlying aim of an expert behind the choice of the task frame is to achieve decoupled control according to two principles: (1) decoupling between the directional and moment component of the motion and wrench screws, and (2) decoupling between motion and wrench.

Principle 1: Decoupled Screw Direction and Moment

Decoupled control between the directional component $\boldsymbol{a}$ and moment component $\boldsymbol{b}$ of a screw, where the screw is either a twist or a wrench, can be achieved by choosing the reference point for the moment component of the screw to be on the line of the screw axis itself. This reference point is then taken as the origin of the task frame. The advantage is that control corrections in the directional component of the screw performed at this point will not affect the moment component of the screw.

Following Principle 1, we propose an approach to automatically derive the origin of the task frame with its viewpoint. This approach starts from two ideal models in which a perfect decoupling can be achieved and determines which ideal model best fits the motion and the wrench data.

We start by defining the two ideal models for the motion:

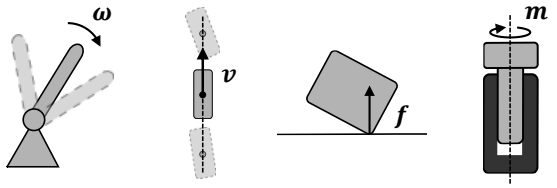


Figure 3.5: Four models with a perfect decoupling between the direction and moment components of a screw: (a) pure rotation about fixed point; (b) constant translation of point; (c) pure force through fixed point; (d) constant moment through fixed point.

Model 1- Pure rotation about fixed point (Figure 3.5a): In this model, a rigid body rotates about a point (e.g. a spherical joint) or about a line (e.g. a hinge), i.e. performing a pure rotation. By choosing the origin in that point or on that line, the translational velocity will be zero. Errors in the orientation can therefore be controlled without affecting the translation.

Model 2- Constant translation of point (Figure 3.5b): In this model, a rigid body performs a constant translation together with a general rotation about a reference point on the body. By choosing the origin in that point on the body, correcting an error in orientation about that point will not affect the constant translational motion of the point.

For the wrench, two similar models are defined:

Model 1- Pure force through fixed point (Figure 3.5c): In this model, the contact wrench between two rigid bodies consists of a pure force through a point (e.g in a point-plane contact), i.e., the resulting contact moment is zero in this point. By choosing the origin in that point the moment will be zero. Correcting an error in force through that point does not affect the moment about the point.

Model 2- Constant moment through fixed point (Figure 3.5d): In this model, the contact wrench between two rigid bodies consists of a constant moment together with a general force through a given point on the body. By choosing the origin in that point, correcting an error in force through that point does not affect the constant moment about the point.

We then apply the ASIP procedure introduced in Section 3.3.6 to both the twist and the wrench screws to find out which of Models 1 and 2 best fits these data. For Model 1, the ASIP procedure is directly applied to screws $\boldsymbol{\mathcal{s}} = \boldsymbol{t}$ or $\boldsymbol{w}$. This results in the point with minimum average velocity or minimum average

moment, respectively. In the ideal case where this minimum is zero, the motion and wrench data exactly correspond to their respective Model 1. In practice, this will never be the case, but the ASIP procedure results in the best possible origins to explain the motion and wrench data under the assumption of Model 1, together with their corresponding covariance matrices (3.22).

For Model 2, the ASIP procedure is applied to the twist and wrench screws after subtracting their average:

$$\breve{\boldsymbol{\jmath}} = \boldsymbol{\jmath} - \text{mean}(\boldsymbol{\jmath}) \tag{3.26}$$

This eliminates the constant moment component of the screw; hence the remaining screw only has a directional component (with the average subtracted), and hence corresponds to the ideal case of Model 1. Consequently, this ASIP procedure results in the point with the most constant velocity or most constant moment, for motion and wrench respectively. In the ideal case where the deviation from a constant velocity or moment is zero, the motion and wrench data exactly correspond to Model 2. In practice this will never be the case, but the ASIP procedure results in the best possible origins to explain both motion and wrench data under the assumption of Model 2, together with their corresponding covariance matrices.

The criterion to decide which of Model 1 and 2 best fits the data is the comparison between the uncertainty (i.e., model misfit) of the two obtained origins, as given by the determinant of their covariance matrices, $\det(\boldsymbol{C}_{asip_1})$ and $\det(\boldsymbol{C}_{asip_2})$.

Section 3.4.3 provides more details about the derivation of the task frame origin.

As an additional result of this procedure, we can select a suitable progress rate to model the task: if Model 1 yields the best fit for the motion data, then the progress rate is best chosen as the rotational velocity, because the translational velocity of the origin of the task frame can become very small, hence is not suited; if Model 2 yields the best fit, then the progress rate is best chosen as the translational velocity of the origin, because this point has the most constant translational velocity.

Principle 2: Decoupled Motion and Wrench

Decoupled control of motion and wrench can be achieved by choosing the orientation for the task frame such that there is either motion or wrench control along and about the orthogonal axes of the frame.

Historically, this approach received a lot of attention in the context of *hybrid motion-force control*, which was conceived for *ideal contacts*, i.e., frictionless

contacts between rigid objects. However, many practical tasks in contact involve non-ideal contacts, and hence other controllers have been devised, such as admittance and impedance controllers, which combine motion and wrench control in the same direction. In our view, this lessens the need for separate control types in different directions. Still, even for these controllers a well-chosen orientation of the task frame can be helpful to specify possibly varying stiffness and damping matrices with different values in different directions.

Following Principle 2, we propose to derive the optimal orientation of the task frame with its viewpoint by analyzing the main direction and its variation of different 3-vector candidates $\boldsymbol{c}$ using the AVOF procedure introduced in Section 3.3.5. Possible inputs to the AVOF procedure are rotational velocity $\boldsymbol{\omega}$, force $\boldsymbol{f}$, translational velocity $\boldsymbol{v}$ and moment of force $\boldsymbol{m}$. The latter two have to be expressed at the origin resulting from Principle 1. From these four candidates we can select suitable *vectors of interest* following the line of reasoning of Principle 1: if Model 1 yields the best fit to the motion or wrench data, then it is not very relevant to analyze the orientation of the translational velocity or moment of force and their variation, because they are very small at the origin, hence we prefer to analyze the rotational velocity or force. Conversely, if Model 2 yields the best fit, then it is more relevant to analyze the orientation of the (almost) constant translational velocity or moment and its variation, and derive the orientation of the task frame accordingly. This is summarized in Table 3.1.

Section 3.4.4 provides more details about the derivation of the task frame orientation.

Table 3.1: Vectors of interest for determination of orientation and geometric progress variable.

Data	**Criterion**	**Model**	**Vectors of Interest (Progress Variable ξ)**
Twist	$\det(\boldsymbol{C}_{asip_1}) < \det(\boldsymbol{C}_{asip_2})$	1: min. $\boldsymbol{v}$	$\boldsymbol{\omega}$ ($\dot{\xi} := \|\boldsymbol{\omega}\|$)
	$\det(\boldsymbol{C}_{asip_1}) > \det(\boldsymbol{C}_{asip_2})$	2: const. $\boldsymbol{v}$	$\boldsymbol{v}$ ($\dot{\xi} := \|\boldsymbol{v}\|$)
Wrench	$\det(\boldsymbol{C}_{asip_1}) < \det(\boldsymbol{C}_{asip_2})$	1: min. $\boldsymbol{m}$	$\boldsymbol{f}$
	$\det(\boldsymbol{C}_{asip_1}) > \det(\boldsymbol{C}_{asip_2})$	2: const. $\boldsymbol{m}$	$\boldsymbol{m}$

3.4.3 Selection of Origin (Figure 3.6a)

As explained in Section 3.4.2 origin candidates for the task frame are found by applying the ASIP procedure to both twist and wrench data, and in two ways: to $\boldsymbol{\mathcal{s}}$ and to $\breve{\boldsymbol{\mathcal{s}}}$. In addition, to be able to decide about the viewpoint for the origin, we have to perform this procedure for screws that are expressed

with respect to both the world frame $\{w\}$ and to the tool frame $\{tl\}$. Hence, in total eight applications of the ASIP procedure are required: to ${}_*\boldsymbol{\mathcal{I}}$ or ${}_*\breve{\boldsymbol{\mathcal{I}}}$, where $\boldsymbol{\mathcal{I}} = \boldsymbol{\mathcal{t}}$ or $\boldsymbol{\mathcal{w}}$, and '$*$' stands for either w or tl. This yields eight origin candidates $\boldsymbol{p}_{asip}$ with their covariance matrix $\boldsymbol{C}_{asip}$. The eight candidates are reduced to a single origin in three steps. First, for each combination of a screw $\boldsymbol{\mathcal{I}}$ and its mean-subtracted counterpart $\breve{\boldsymbol{\mathcal{I}}}$, we select the origin with the least uncertainty, i.e. with the smallest $\det(\boldsymbol{C}_{asip})$. Note that the unit of this uncertainty metric is always $[\mathrm{m}^6]$ and that this step boils down to selecting between Model 1 or Model 2 for both twist and wrench screws in the two viewpoints. Next, since the uniform metric allows us to combine the results of twist and wrench analyses, the remaining four origin candidates are averaged two by two in each viewpoint, weighted by the inverse of their covariance matrix. I.e., given candidates $\boldsymbol{p}_{asip_t}$ and $\boldsymbol{p}_{asip_w}$ with corresponding $\boldsymbol{C}_{asip_t}$ and $\boldsymbol{C}_{asip_w}$, the averaged origin and its covariance matrix correspond to:

$$\boldsymbol{p}_{avg} = \boldsymbol{C}_{avg}\left(\boldsymbol{C}_{asip_t}^{-1}\ \boldsymbol{p}_{asip_t} + \boldsymbol{C}_{asip_w}^{-1}\ \boldsymbol{p}_{asip_w}\right), \tag{3.27}$$

and:

$$\boldsymbol{C}_{avg}^{-1} = \boldsymbol{C}_{asip_t}^{-1} + \boldsymbol{C}_{asip_w}^{-1}. \tag{3.28}$$

Finally, to select the least uncertain viewpoint, $\det(\boldsymbol{C}_{avg})$ is again used to decide between the averaged origins in the two viewpoints.

So, the outcomes of this procedure are a viewpoint for the origin, a value for the origin and its covariance matrix. Backtracking to the two selections made between the ASIP results in box a-4 for the selected viewpoint, we can further decide about the selection of the progress rate $\dot{\xi}$ and about the selection of the vectors of interest for the determination of a suitable orientation of the task frame, according to Table 3.1. No thresholds or other parameters are involved in the whole procedure.

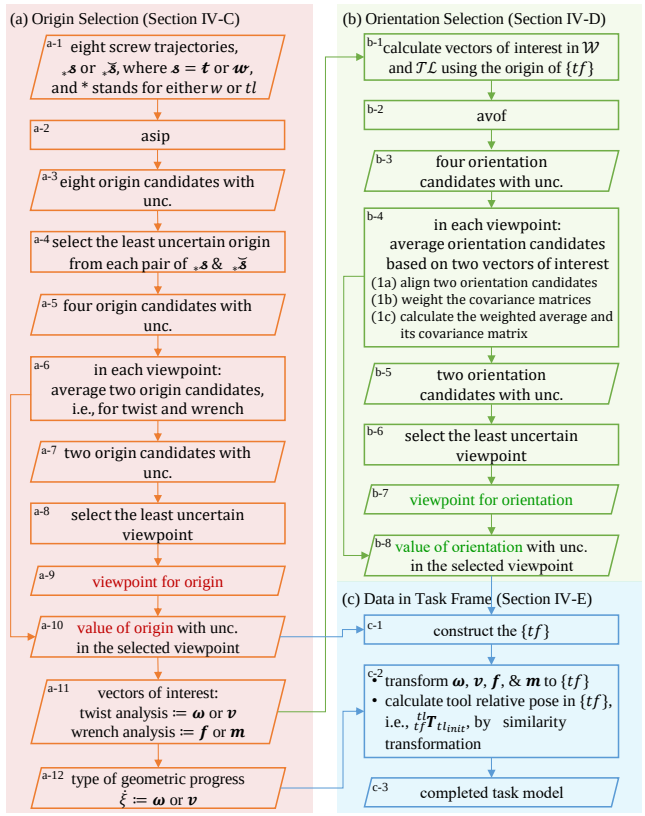


Figure 3.6: Flow chart of the proposed approach. In this figure, unc. stands for uncertainty. (a) The origin of the task frame is derived, explained in Section 3.4.3. (b) The orientation of the task frame is derived, detailed in Section 3.4.4. (c) The demonstration data including pose, twist, and wrench are expressed in the derived task frame as elaborated in Section 3.4.5.

3.4.4 Selection of Orientation (Figure 3.6b)

As explained in Section 3.4.2 orientation candidates for the task frame are found by applying the AVOF procedure to the vectors of interest which result from the origin selection procedure in Section 3.4.3: either $\boldsymbol{\omega}$ or $\boldsymbol{v}$ for twist, and either $\boldsymbol{f}$ or $\boldsymbol{m}$ for wrench. Values of $\boldsymbol{v}$ and $\boldsymbol{m}$ need to correspond to the translational velocity and moment of force at the selected task frame origin. This change to reference point tf is achieved using (3.14). In addition, to be able to decide about the viewpoint for the orientation, we have to perform this

procedure twice, once with all vectors expressed with respect to $\{w\}$ and once with all vectors expressed with respect to $\{tl\}$. Hence, four applications of the AVOF procedure are required: to ${}_{\circ}\boldsymbol{c}$, where $\boldsymbol{c}$ is either $\boldsymbol{\omega}$ or $\boldsymbol{v}$, and either $\boldsymbol{f}$ or $\boldsymbol{m}$, and '$\circ$' stands for w or tl.

The AVOF procedure yields four orientation candidates $\boldsymbol{U}_{avof}$ with their covariance matrix $\boldsymbol{C}_{avof}$ as found from (3.18) and (3.19), respectively. These four candidates are reduced to a single orientation in two steps: (1) averaging the two candidates resulting from either $\boldsymbol{\omega}$ or $\boldsymbol{v}$ and either $\boldsymbol{f}$ or $\boldsymbol{m}$ in every viewpoint and (2) selecting the viewpoint with the least resulting uncertainty. The first step is further divided into three substeps: (1a) aligning the two orientation candidates, (1b) weighting the covariance matrices, and (1c) calculating the weighted average and its covariance matrix. These substeps are summarized below.

(1a) Aligning two orientation candidates. In every viewpoint, the two AVOF procedures result in two rotation matrices, $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$, from twist and wrench data, respectively. The purpose is to find the best correspondence between the columns of $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ and label the corresponding pairs with x, y or z such that we obtain two rotation matrices $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ that are more or less aligned, as illustrated in Figure 3.7a, 3.7b.

Without loss of generality we choose the first orientation candidate as $\boldsymbol{R}_1 = \boldsymbol{U}_1$. Consequently, its x-axis is chosen according to the direction of the average $\boldsymbol{\omega}$ or $\boldsymbol{v}$, i.e. the main motion direction. Then the columns of $\boldsymbol{U}_2$ are reordered while assigning a proper sign to them to maintain righthandedness, such that the resulting orthonormal matrix corresponds to the rotation matrix $\boldsymbol{R}_2$ with the closest distance to $\boldsymbol{R}_1$. Algorithm 3 provides the detailed procedure. Note that this procedure also supports the case where the average direction of $\boldsymbol{f}$ or $\boldsymbol{m}$ is (almost) parallel to $\boldsymbol{v}$ or $\boldsymbol{\omega}$, i.e. also along the x-axis, as opposed to (almost) orthogonal as in the hybrid control paradigm. Also, this procedure does not affect $\boldsymbol{C}_{avof_2}$, because it remains defined by the corresponding input vectors $\boldsymbol{c}$, which are still expressed with respect to the same reference frame.

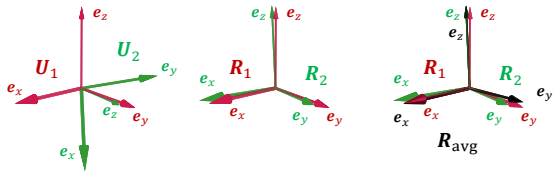


Figure 3.7: Role of alignment and calculation of weighted average of two orientation frames $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$. (a) The two orientation frames $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$. (b) The two orientation frames after alignment resulting in $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$. (c) The average of $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ is shown in black.

Algorithm 3: Align two sets of three orthonormal vectors resulting in two orientation frames with minimum distance

Input: $\boldsymbol{U}_1, \boldsymbol{U}_2$
Output: $\boldsymbol{R}_1, \boldsymbol{R}_2$

$\boldsymbol{R}_1 \leftarrow \boldsymbol{U}_1$;
$\boldsymbol{R}_\Delta \leftarrow \boldsymbol{U}_2^T \, \boldsymbol{R}_1$;
// c is a column index , r is a row index
// $\boldsymbol{P}$ permutates and adapts signs of columns.
for $c = 1 \dots 3$ **do**
    $r \leftarrow \arg\max \;\; \text{abs}\left(\boldsymbol{R}_\Delta(:, c)\right)$;
    $\boldsymbol{P}(:, c) \leftarrow 0$;
    $\boldsymbol{P}(r, c) = \text{sign}(\boldsymbol{R}_\Delta(r, c))$;
    // ensure that row cannot be selected in subsequent columns:
    $\boldsymbol{R}_\Delta(r, :) \leftarrow 0$;
end
$\boldsymbol{R}_2 = \boldsymbol{U}_2 \boldsymbol{P}$;

(1b) Weighting the nondimensional covariance matrices. Using the nondimensional $\boldsymbol{C}_{avof_1}$ and $\boldsymbol{C}_{avof_2}$ resulting from (3.19) to calculate the average orientation of $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ has a benefit: it is completely data-driven, hence it does not involve any hyperparameters. However, the downside is that both rotation matrices are weighted according to how strong the vectors of interest $\boldsymbol{c}$ vary with respect to their average magnitude, which is not desired if this average magnitude is very small. An example is: very small, hence negligible contact forces measured in a human demonstration. In such case, the obtained orientation of these forces is irrelevant for the task model and should be discarded to avoid undesired bias in the averaged orientation. This can be achieved in an ad-hoc way based on an expert's insight, or automatically by applying an *additional* weighting, i.e. after applying (3.19). One approach is to choose the

weights according to:

$$\boldsymbol{C}_{avof} \leftarrow \boldsymbol{C}_{avof} \frac{c_{ref}^2}{c_{avgsn}} \tag{3.29}$$

where c_{ref} is a reference value for the magnitude of $\boldsymbol{c}$ below which, for example, the magnitude of the vector of interest is considered to be irrelevant, while c_{avgsn} is the average of the squared norm of $\boldsymbol{c}$, which was also used in (3.19). Such reference values are chosen for each of $\boldsymbol{\omega}$, $\boldsymbol{v}$, $\boldsymbol{f}$ and $\boldsymbol{m}$, which introduces four hyperparameters.

(1c) Calculating the weighted average. Given $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ with their resulting nondimensional and (optionally) weighted $\boldsymbol{C}_{avof_1}$ and $\boldsymbol{C}_{avof_2}$, the averaged orientation $\boldsymbol{R}_{avg}$ shown in black in Figure 3.7c can be found using the iterative procedure in Algorithm 4. This algorithm was first proposed for quaternions and dual quaternions in [54] but was adapted here to rotation matrices. See Appendix C for the details. The covariance of $\boldsymbol{R}_{avg}$ is calculated according to (C.9).

Finally, the viewpoint for the orientation is selected by comparing the determinant of the resulting covariance matrix and choosing the viewpoint with the smallest determinant.

So, the outcomes of this procedure are a viewpoint for the orientation, a value for the orientation and its covariance matrix. If there is no additional weighting in step (1b), no hyperparameters are involved in the whole procedure. Additional weighting introduces four hyperparameters, but they can be chosen intuitively and identical for a class of comparable applications.

Algorithm 4: Weighted average of two rotations

Input: $\boldsymbol{R}_1$, $\boldsymbol{R}_2$, $\boldsymbol{C}_1$, $\boldsymbol{C}_2$
Output: $\boldsymbol{R}_{avg}$

$\boldsymbol{R} \leftarrow \boldsymbol{R}_1$; `// initialize`
$\boldsymbol{\Lambda}_1 \leftarrow \left(\boldsymbol{C}_1^{-1} + \boldsymbol{C}_2^{-1}\right)^{-1} \boldsymbol{C}_1^{-1}$; `// weight 1`
$\boldsymbol{\Lambda}_2 \leftarrow \left(\boldsymbol{C}_1^{-1} + \boldsymbol{C}_2^{-1}\right)^{-1} \boldsymbol{C}_2^{-1}$; `// weight 2`
repeat
  $\boldsymbol{\delta} \leftarrow \boldsymbol{\Lambda}_1 \log^{\vee}(\boldsymbol{R}_1 \boldsymbol{R}^{-1}) + \boldsymbol{\Lambda}_2 \log^{\vee}(\boldsymbol{R}_2 \boldsymbol{R}^{-1})$;
  $\boldsymbol{R} \leftarrow \exp(\boldsymbol{\delta}) \boldsymbol{R}$; `// update rotation`
until $\|\boldsymbol{\delta}\| < \epsilon$;
$\boldsymbol{R}_{avg} \leftarrow \boldsymbol{R}$;

3.4.5 Representation of Data in the Task Frame (Figure 3.6c)

From the origin ${}^{tf}\boldsymbol{p}_*$ and orientation ${}^{tf}\boldsymbol{R}_\circ$ of the task frame with their respective viewpoints, we can construct a transformation matrix for the task frame with

respect to a general reference frame $\{ref\}$:

$$ {}^{tf}\boldsymbol{T}_{ref} = \begin{bmatrix} {}^{\circ}\boldsymbol{R}_{ref}\, {}^{tf}\boldsymbol{R}_{\circ} & {}^{*}\boldsymbol{p}_{ref} + {}^{*}\boldsymbol{R}_{ref}\, {}^{tf}\boldsymbol{p}_{*} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \tag{3.30} $$

In the robot controller, typically ${}^{tf}\boldsymbol{T}_{w}$ or ${}^{tf}\boldsymbol{T}_{tl}$ are needed, i.e. with $ref = w$ or tl. Obviously, these matrices are only constant if the viewpoints of both origin and orientation correspond to ref. In other cases, the transformation matrix changes with the progress of the task. For example, for $ref = tl$, $* = tl$ and $\circ = w$, (3.30) becomes:

$$ {}^{tf}\boldsymbol{T}_{tl} = \begin{bmatrix} {}^{w}\boldsymbol{R}_{tl}\, {}^{tf}\boldsymbol{R}_{w} & +{}^{tf}\boldsymbol{p}_{tl} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \tag{3.31} $$

where ${}^{w}\boldsymbol{R}_{tl} = {}^{tl}\boldsymbol{R}_{w}^{-1}$ is known from the instantaneous robot kinematics.

Then, we express all the demonstration data in $\{tf\}$. For twist $\boldsymbol{t}$ and wrench $\boldsymbol{w}$ this is achieved by transforming these screws to ${}_{tf}\boldsymbol{\jmath}$ according to (3.13). For pose, the data is given by ${}^{tl}\boldsymbol{T}_{w}$ and can be decomposed into the initial pose of the tool, ${}^{tl_{init}}\boldsymbol{T}_{w}$, and the tool trajectory relative to that initial pose:

$$ {}^{tl}\boldsymbol{T}_{tl_{init}} = {}^{tl_{init}}\boldsymbol{T}_{w}^{-1}\, {}^{tl}\boldsymbol{T}_{w} \tag{3.32} $$

By storing only the relative tool trajectory, the pose data becomes independent of the initial pose. This is very useful for transferring the skill from the demonstration environment to the robot, because usually the robot environment is different from the demonstration environment. For example, the world frame $\{w\}$ will typically not be at the camera or other sensor as during the demonstration, but instead at the robot base. Even if a model of this transformation is known, there might be calibration errors. Also, the object or workpiece that is fixed to $\{w\}$ could be placed at another location. All of this is taken care of by this approach. As an additional step, to be able to use the pose data in a controller that operates in the task frame, the pose trajectory ${}^{tl}\boldsymbol{T}_{tl_{init}}$ is expressed in the task frame using similarity transformation (3.6):

$$ {}^{tl}_{tf}\boldsymbol{T}_{tl_{init}} = {}^{tf}\boldsymbol{T}_{tl_{init}}^{-1}\; {}^{tl}\boldsymbol{T}_{tl_{init}}\; {}^{tf}\boldsymbol{T}_{tl_{init}} \tag{3.33} $$

where generally ${}^{tf}\boldsymbol{T}_{tl_{init}}$ is a variable matrix given by

$$ {}^{tf}\boldsymbol{T}_{tl_{init}} = {}^{tl_{init}}\boldsymbol{T}_{w}^{-1}\, {}^{tf}\boldsymbol{T}_{w} \tag{3.34} $$

with ${}^{tf}\boldsymbol{T}_{w}$ obtained from (3.30).

So, the demonstration data is represented as ${}_{tf}\boldsymbol{t}$, ${}_{tf}\boldsymbol{w}$ and ${}^{tl}_{tf}\boldsymbol{T}_{tl_{init}}$. This completes the task frame derivation procedure. The demonstration data can

be further processed, for example to align different demonstration trials by expressing them as a function of geometric progress ξ, and analyzed, for example by making statistical models based on the trials. This can be done in multiple ways, as in [53],[29], and [102], and is out of scope of this chapter.

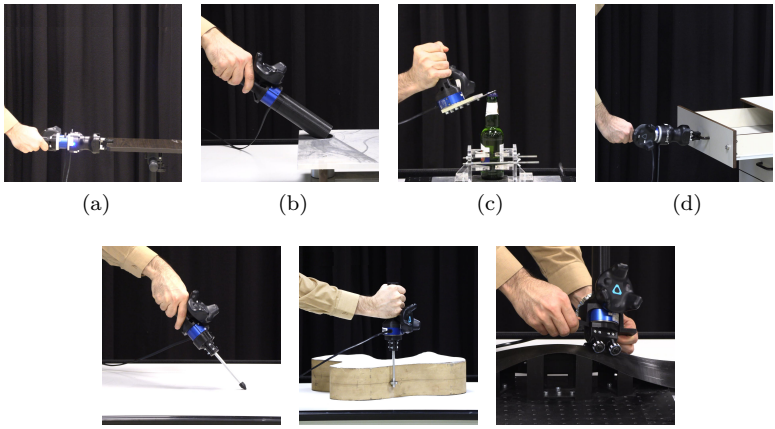
3.5 Experiments Design

To validate the task frame derivation approach, it is subjected to a diverse range of applications. We anticipate that the approach will produce results closely, but not necessarily perfectly, aligned with expert choices. This expectation arises from the fact that in complex tasks, experts may rely on best guesses grounded in simplifying assumptions, and the algorithm's solution may surpass the expert's choice. We also want to investigate the effect of the optional weighting in the derivation of the task frame's orientation.

To validate the benefits of using a well-chosen and parameterized task model, the task model obtained in Section 3.4.5 is utilized to perform demonstrated tasks by a robot. In pursuit of this, a general-purpose controller is designed based on the derived task frame. The controller's effectiveness is validated in the various applications. To assess its robustness and generalization capabilities, diverse scenarios involving disturbances or parameter variations are considered. To affirm the significance of utilizing a well-chosen task frame we investigate whether alterations in the task frame result in a deterioration of task performance. Additionally, controlling tasks within the derived task frame serves to validate that the demonstration data (twist, pose and wrench), represented as explained in Section 3.4.5, is accurate and sufficient, enabling accurate task performance by the robot.

3.5.1 Applications

Figure 3.8 depicts the seven tasks involving motion in contact considered in the experiments. Below we provide further explanation for each of them, paying particular attention to the task variability that was present in the demonstrations, which always consisted of five trials.



(a) (b) (c) (d)

(e) (f) (g)

Figure 3.8: Setups used for teaching different tasks to the robot: (a) Revolute Joint (b) Peg on Hole Alignment (c) Bottle Opening (d) Prismatic Joint (e) Drawing (f) 2-D Contour Following (g) 3-D Contour Following. The pictures are snapshots of the videos in the supplementary material. The demonstration data for Bottle Opening and 2-D Contour Following are borrowed from [103].

Revolute Joint

Interacting with a revolute joint such as when opening a door is a common daily activity. This task has one DOF for motion space and five DOFs for wrench space. The tool is equipped with a gripper that can grip one body attached to the joint, in our case a table, and rotate it around the hinge. The setup is shown in Figure 3.8a. The operator demonstrated the task in different positions and with the joint in different horizontal orientations with respect to the world frame. The amount of rotation was almost the same in the different trials.

Peg on Hole Alignment

This task can be used as an initialization before peg insertion in case it is difficult to provide a good estimate of the hole's location beforehand. First, we establish a three-point contact between the intentionally misaligned peg and the hole. Then, we rotate the peg until the symmetry axis of the peg is aligned with the symmetry axis of the hole. Unlike a simple revolute joint, this alignment involves a moving rotation axis as shown in [25]. This task has three DOFs for both motion and wrench spaces. The peg is mounted on the tool and

the hole is attached to the environment as shown in Figure 3.8b. The operator demonstrated the task in the same location and orientation with respect to the world frame. However, slight differences in the tool's orientation among demonstrations are observed due to human variability. The amount of initial misalignment is nearly the same among trials.

Bottle Opening

Removing the cap from a bottle is a non-reciprocal task, i.e., we need to apply moment and rotate the bottle opener in the same direction to remove the cap. Hence, unlike the previous tasks, this one requires non-negligible power consumption. An opener is mounted on the tool and a bottle is fixed to the environment as shown in Fig 3.8c. At the start of the task, initial contact is well-established, with the opener completely locked. Hence, there are zero and six DOFs in the motion and wrench spaces, respectively. During the opening phase, application of the interaction wrench causes motion, mainly but not exclusively in rotation. The demonstration data for this application is borrowed from [103]. The operator demonstrated the task in different positions, with slightly different orientations of the bottle's symmetry axis with respect to the vertical axis of the world frame. The amount of rotation is consistent among trials.

Prismatic Joint

Interacting with a prismatic joint such as when opening a drawer is another common daily activity. This task has one DOF for motion space and five DOFs for wrench space. The same tool employed for interacting with a Revolute Joint was used, as shown in Figure 3.8d. The operator always demonstrated the task in the same position and orientation with respect to the world frame, with almost the same amount of translation.

Drawing

Tasks like drawing, writing and pizza cutting with a roller all involve a point-on-plane contact. The objective is to exert a force normal to a surface such as a table and simultaneously translate the tool over the table. In this study we considered drawing. This task has one DOF for wrench space and five DOFs for motion space. While the two translational DOFs along the table are functional during the demonstrations, the three orientations can be varied freely by the operator. The tool is equipped with a pen-like rod as shown in Figure 3.8e.

The operator demonstrated the task in the same location and orientation with respect to the world frame. There is not much variation in the tool's orientation with respect to the world frame and the plane of drawing remained the same among the trials.

2-D Contour Following

The objective of this task, shown in Figure 3.8f, is to follow a 2-D contour using a roller at almost constant translational velocity while maintaining an almost constant contact force. Polishing is a typical application. The task has only one DOF for wrench space, while there are five DOFs for motion space. One translational DOF along the contour allows us to follow it. Since it is a 2-DOF application, the tool is expected not to move in the two out-of-plane rotational directions and the out-of-plane translational direction, hence maintaining a constant orientation or position in those directions. Finally, the intention was to maintain also a constant in-plane orientation of the tool with respect to the world. The tool consists of a rod with a roller at its tip. The demonstration data was again borrowed from [103]. The operator demonstrated the task in different locations but with approximately the same orientation of the workpiece with respect to the world frame. Hence, also the plane of the contour remained the same among the trials.

3-D Contour Following

The objective is to follow a 3-D contour with almost constant translational velocity while simultaneously exerting an almost constant amount of force normal to the contour's edge. The tool is shown in Figure 3.8g, see [108] for more details. It has five wheels which are expected to be in contact during the experiment. The loss of contact for any of these wheels is considered a task failure. The task involves one instantaneous DOF in motion space, i.e. translation along the contour's edge, and five instantaneous DOFs in wrench space, i.e. normal forces on two sides of the contour and three moments. Compared to the previous tasks, this task required much more skill from the operator to complete successful demonstrations, as it was not evident to maintain contact at all wheels during the entire trials. The operator demonstrated the task with the workpiece in the same position and orientation with respect to the world frame.

3.5.2 Demonstration Setup

The HTC VIVE system was used for recording pose data in all applications. This system includes a tracker that was mounted on the tool and a camera for recording the tracker's pose. Its accuracy is in the order of a few millimeters and a few degrees. In contrast, different force/torque sensors were used in the various applications, as detailed in Table 3.2 along with their data sheet specifications. The first three sensors were used for recording demonstration data (respectively for applications 1/2/4/5, 3/6 and 7), while the last one was employed for all tasks performed by the robot. All the sensors used in the demonstration setups record data at a frequency of 200 Hz. Furthermore, in applications 1 and 4, we used the Robotiq Hand-E gripper; it has two states: open and close.

Table 3.2: Specifications of the force/torque sensors.

6-Axis Force/Torque sensor	**Nominal Accuracy**	**Measurement Range**		
		Force [N]		**Torque** [Nm]
		f_x, f_y	f_z	τ_x, τ_y, τ_z
JR3 50M31A3	1%	±100	±200	±5
JR3 67M25A3	1%	±200	±400	±12
JR3 67M25T3	1%	±400	±800	±24
Schunk FTN-AXIA80	< 2%	±150	±470	±8

3.5.3 Pre-Processing of Demonstration Data

The motion tracker records the pose of a frame $\{tr\}$ attached to the tracker with respect to the camera frame $\{c\}$, which corresponds to the world frame $\{w\}$. Simultaneously, a force/torque sensor records wrench measurements in the force/torque sensor frame $\{fs\}$. Both tracker and force/torque sensor are attached to the tool frame $\{tl\}$. Their relative poses were assumed to be known.

To start, we segmented the data to identify the motion-in-contact part of each demonstration trial. We opted for a straightforward approach based on signal magnitude, since segmentation was not the focus of this study. A first segmentation was performed based on the magnitude of the force $\boldsymbol{f}$ or moment $\boldsymbol{m}$ to make sure there was contact. This implies prior weight compensation of the measured wrench. We additionally processed the wrench signals using a moving average smoother weighted with a Gaussian distribution. Subsequently, we further segmented the data based on the magnitude of the rotational velocity $\boldsymbol{\omega}$ or translational velocity $\boldsymbol{v}$ to eliminate the parts without significant motion.

The pose data ${}^{tl}\boldsymbol{T}_w$ used in the task frame derivation procedure was then calculated from the measured tracker poses ${}^{tr}\boldsymbol{T}_w$ and the relative pose ${}^{tr}\boldsymbol{T}_{tl}$, while the twist data ${}_w\boldsymbol{t}$ followed from pose differentiation according to (3.11). Next, we augmented the five demonstration trials for each application to obtain a batch. Lastly, we calculated screws $\boldsymbol{\mathfrak{z}}$ and $\breve{\boldsymbol{\mathfrak{z}}}$, cf. (3.26), in both viewpoints as required for origin selection in Figure 3.6.

3.5.4 Task Frame Derivation and Data Post-Processing

Task frame derivation was applied according to Section 3.4 for all applications. No regularization was applied in the ASIP procedure, i.e. ϵ was set to zero in (3.21). Furthermore, following reference values were selected for the optional weighting of the orientation: $\omega_{ref} = 0.05$ rad/s, $v_{ref} = 0.005$ m/s, $f_{ref} = 1$ N, and $m_{ref} = 0.1$ Nm. No further tuning of these values was applied.

After transforming the data to the obtained task frame according to Section 3.4.5, the following signals are determined: x, y, z-positions and unit quaternions for pose, and the components of the twist and wrench screws. The signals were made time-invariant and aligned to handle differences in velocity profile by reparameterizing them according to (3.15) as functions of the nondimensional geometric progress variable $\bar{\xi}$ that resulted from the task frame derivation, while using the same number of samples $N = 100$ for all five trials. Next, we fitted a cubic spline to these aligned signals to find a smooth average over the trials. The quaternions are renormalized after smoothing.

In a final step, $\bar{\xi}$ was replaced by $\bar{\xi}.\xi_{max,avg}$ where $\xi_{max,avg}$ is the average traveled distance or angle of the trials. The resulting signals were saved as the desired reference signals expressed in the task frame.

3.5.5 Task Performance Setup and Control Strategy

A UR10e robot is used to perform the tasks. The robot is equipped with a Schunk force/torque sensor, see Table 3.2 for its specifications, and is controlled with a frequency of 100 Hz. We employed a constraint-based and velocity-resolved control framework known as *eTaSL* [9], and devised a single type of controller for all applications. Evidently, other types of controllers can be applied as well. In this approach desired values for pose, twist and wrench are combined in two sets of constraints that are applied simultaneously, as detailed below.

Pose constraints are responsible for tracking the desired pose trajectory expressed in the task frame, ${}^{tl}_{tf}\boldsymbol{T}^{\ des}_{tl_{init}}$. The actual pose trajectory is also expressed in the task frame through a similarity transformation applied to the relative trajectory with respect to the initial pose, ${}^{tl}\boldsymbol{T}^{act}_{tl_{init}}$, in the same way as the desired trajectory in (3.33), but now in real-time:

$$ {}^{tl}_{tf}\boldsymbol{T}^{act}_{tl_{init}} = \left({}^{tf}\boldsymbol{T}^{act}_{tl_{init}}\right)^{-1} \ {}^{tl}\boldsymbol{T}^{act}_{tl_{init}} \ \ {}^{tf}\boldsymbol{T}^{act}_{tl_{init}} \tag{3.35} $$

The pose constraint ensures that the pose difference:

$$ \Delta\boldsymbol{T} = \left({}^{tl}_{tf}\boldsymbol{T}^{act}_{tl_{init}}\right)^{-1} \ {}^{tl}_{tf}\boldsymbol{T}^{des}_{tl_{init}} \tag{3.36} $$

evolves to the identity matrix $\boldsymbol{I}_4$ using a desired control twist:

$$ {}_{tf}\boldsymbol{t}_{contr,p} = \ k_p \log^{\vee}\left(\Delta\boldsymbol{T}\right) + {}_{tf}\boldsymbol{t}_{des} \tag{3.37} $$

Here, the first term is the feedback term, comprising feedback gain k_p multiplied by the displacement screw derived from the pose error $\Delta\boldsymbol{T}$ according to (3.9). The second term is a feedforward term which aims at reducing the reference tracking error.

Wrench constraints ensure a safe interaction between the tool and the environment and aim at tracking the desired wrench trajectories. This is expressed as a constraint, ${}_{tf}\boldsymbol{w}_{act} = {}_{tf}\boldsymbol{w}_{des}$, which is also converted to a desired control twist:

$$ {}_{tf}\boldsymbol{t}_{contr,w} = \ k_w\hat{\boldsymbol{C}}_w({}_{tf}\boldsymbol{w}_{des} - {}_{tf}\boldsymbol{w}_{act}) + {}_{tf}\boldsymbol{t}_{des} \tag{3.38} $$

where k_w is a feedback gain and $\underline{\Delta}\,\hat{\boldsymbol{C}}_w$ is a diagonal compliance matrix, where $\underline{\Delta} = \begin{bmatrix} \boldsymbol{0}_3 & \boldsymbol{I}_3 \\ \boldsymbol{I}_3 & \boldsymbol{0}_3 \end{bmatrix}$. They are multiplied by the wrench error to form the feedback term. The second term in (3.38) is the same feedforward term as in (3.37).

The feedforward terms in (3.37) and (3.38) generate the exact required motion if the demonstration and robot scenarios are exactly the same. Hence, little or no contribution from the feedback terms is required in that case. If however the robot scenario deviates from the demonstration scenario, the feedback terms in the pose and wrench constraints come into play to compensate for these deviations. As a result, the desired control twists (3.37) and (3.38) are no longer consistent, but conflicting. To solve this conflict, weights w_p and w_w are applied to these equations, respectively, which results in a spring-damper like behavior, as depicted in Figure 3.9. The equivalent stiffness and damping parameters, k_{eq} and b_{eq}, are determined by gains k_p and k_w, weights w_p and w_w, and compliance $\underline{\Delta}\,\hat{\boldsymbol{C}}_w$. This relation can be shown for a 1-DOF translational system,

$$ k_{eq} = \frac{w_p k_p}{w_w k_w \hat{C}_f} \ \text{and} \ \ b_{eq} = \frac{1}{w_w k_w \hat{C}_f} \tag{3.39} $$

where $\hat{C}_f$ is the translational compliance. This is explained in more detail in Appendix D.

Given the large number of setups, little effort was spent in parameter tuning. The control parameters used in every application are listed in Table 3.3. The same control gains $k_p = k_w = 3\ \mathrm{s}^{-1}$ were selected for all applications. These gains correspond to the achievable bandwidth expected from closed-loop control with robots like the UR10e, with a safety margin. We roughly estimated the diagonal entries of the compliance matrix $\underline{\Delta}\,\hat{\boldsymbol{C}}_w$ for every application set-up, as a simplified model for the true physical compliance between the tool and the environment, thereby ensuring stable closed-loop wrench control. We took the same value $\hat{C}_f$ for the three translational directions and the same value $\hat{C}_m$ for all three rotational directions. Furthermore, for each application we selected from three possible weights $w_p = 0.01,\ 0.03$ or 0.1, while $w_w = 1 - w_p$. So, only three parameters were selected for each application: rough estimates for $\hat{C}_f$ and $\hat{C}_m$, and one out of three options for w_p.

The selection of weight parameters (w_p and w_w) was based on a trade-off between two limiting cases:
First case, $w_p = 1$ and $w_w = 0$: In this configuration, only pose constraints are applied, preventing the controller from adapting to variations in the environment's geometry. This is undesirable, as some level of geometric uncertainty is almost always present in real-world applications. Therefore, incorporating wrench constraints is essential to enable adaptation to such changes.
Second case, $w_p = 0$ and $w_w = 1$: In this case, only wrench constraints are considered, making the controller vulnerable to disturbances in the wrench measurements. Such disturbances can accumulate over time through integration, leading to unwanted motion deviations in the degrees of freedom where there is no geometric constraint imposed by the environment. As a result, pose constraints are needed to suppress these drifts.
The values of w_p are chosen significantly smaller than those of w_w because the pose constraint is included primarily to limit the robot's motion in directions not constrained by the geometry of the environment. For instance, in the 2-D Contour Following task, the tool's orientation and its vertical position are not geometrically constrained. If we rely solely on the wrench constraint (3.38), with desired values approximately zero along the unconstrained directions, deviations might occur due to the sensor noise. This is investigated in Section 3.6.2. Moreover, it is worth noting that the choice of weights can be fine-tuned, as there exists a direct relation between the estimated stiffness and the weight parameters according to (3.39). Therefore, by adjusting the estimated stiffness, the weights can be unified accordingly. However, since controller design is not the primary focus of this chapter, we proceed using only three distinct weight

configurations.

The performance of the controller proposed above is anticipated to be independent of the task frame orientation, because (1) the postprocessing of the demonstration data was limited to reparameterization and spline fitting, so no orientation-specific data was thrown away, and (2) all controller gains, compliance values and weights are equal in the three translational directions and in the three rotational directions. This will not be the case with more sophisticated controllers which aim at leveraging non-isotropic design. The decoupling introduced in principle 2 of section 3.4.2 can help in designing such controllers. Such controllers are however out of scope of this chapter.

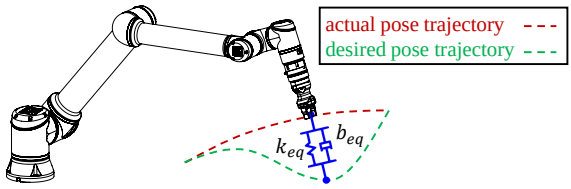


Figure 3.9: Schematic representation of the proposed constraint-based control strategy. Deviation from the desired pose trajectory occurs as a result of the wrench constraint. This can be modeled as a spring-damper system.

Table 3.3: Control parameters used in the experiments.

Task	k_p [s^{-1}]	k_w [s^{-1}]	$\hat{C}_f$ e^{-3}[m/N]	$\hat{C}_m$ e^{-3}[rad/Nm]	w_p [–]	w_w [–]
Revolute Joint	3	3	0.4	0.7	0.01	0.99
Peg on Hole Alignment	3	3	0.7	0.7	0.10	0.90
Bottle Opening	3	3	0.1	100	0.01	0.99
Prismatic Joint	3	3	0.5	0.5	0.03	0.97
Drawing	3	3	0.5	200	0.10	0.90
2-D Contour Following	3	3	0.8	200	0.03	0.97
3-D Contour Following	3	3	0.1	50	0.01	0.99

3.5.6 Scenarios

The proposed control approach is challenged in three types of scenarios, as outlined below.

Nominal scenarios. Here, the robot first performs the task under the same conditions as observed during the demonstrations, i.e. the tool's pose, the contact wrench and the task speed closely resemble what was extracted from the demonstrations. Subsequently, following features were investigated: role of

the pose and wrench constraints, independence of the initial pose, and variation of task speed and contact wrench.

Changed geometry. If the tool is changed from the one used in the demonstrations, this may affect task frame parameters ${}^{tf}\boldsymbol{p}_{tl}$ and/or ${}^{tf}\boldsymbol{R}_{tl}$. Similarly, the geometry of the environment instead of the tool can be different from the one during the demonstration. Two approaches were investigated: (1) changing the task frame parameters manually, based on physical insight, and (2) performing the task with the original parameters, recording the data and reintroducing them in the task frame derivation procedure to find improved parameters.

Sensitivity to task frame origin and orientation. We performed two experiments in which the task frame was changed manually with respect to the optimal one that resulted from the procedure proposed in Section 3.4: a change of orientation and a change of origin.

3.6 Experimental Results

Video recordings of all demonstrations and all robot task performances can be found in the supplementary material.[3]

3.6.1 Task Frame Derivation

Table 3.4 summarizes the results of the task frame derivation. Columns 2-5 contain the results of the discrete decisions: viewpoints for the origin and the orientation ($\mathcal{W}$ or $\mathcal{TL}$), and the vectors of interest for motion and wrench ($\boldsymbol{v}$ or $\boldsymbol{\omega}$, and $\boldsymbol{f}$ or $\boldsymbol{m}$). Each decision is based on the comparison of two relevant covariance matrices $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$. Therefore, between brackets, we also report a ratio,

$$\text{Ratio} = \sqrt{\frac{\max(\det(\boldsymbol{C}_1), \det(\boldsymbol{C}_2))}{\min(\det(\boldsymbol{C}_1), \det(\boldsymbol{C}_2))}}, \tag{3.40}$$

as a measure of the statistical significance of the decision. While a large value of the ratio corresponds to a large significance, a value close to 1 corresponds to a low significance. I.e., in the latter case the other option might work equally well in practice.

Columns 6-13 focus on the values derived for the origin and orientation of the task frame, $\{tf\}$, in comparison with the expert's choice for the task frame,

[3]The Matlab codes and the demonstration data will be available as a Git repository with the final version of the chapter.

$\{tf^*\}$. The expert choice was based on the authors' best judgment. Derived and expert task frames are visualized for each application in the video recordings in the supplementary material. The differences between the origins and the orientations of $\{tf\}$ and $\{tf^*\}$ were calculated in different ways, to capture their most relevant component(s) for each particular application. This is detailed at the bottom of the table. For example, for all applications except Drawing and 2-D Contour Following, the angle between the main axes of motion, ${}^{[tf]}\boldsymbol{e}_x$ and ${}^{[tf^*]}\boldsymbol{e}_x$, was chosen for evaluating the difference in orientation, while for the other two the angle between the normals to the plane of motion, ${}^{[tf]}\boldsymbol{e}_z$ and ${}^{[tf^*]}\boldsymbol{e}_z$, was chosen. The differences in origin and orientation are reported for four task frame candidates: the task frames resulting from the motion data, from the wrench data, their unweighted average and their weighted average. For each application, the smallest differences are marked in bold.

Viewpoint. Viewpoint $\mathcal{TL}$ was consistently selected for the origin in all applications. The decision was particularly significant in all applications where the point of contact moved with respect to the world, for example in the Drawing task where the contact force is always applied at the tip of the tool, or which were demonstrated in different positions with respect to the world, for example in the Bottle Opening. Conversely, in applications such as Peg on Hole Alignment and Prismatic Joint, where the contact point does not move a lot with respect to the world, and/or which were always demonstrated in the same position with respect to the world, the decision had a low significance. When analyzing the individual trials of the Peg on Hole Alignment we even observed that $\mathcal{TL}$ was favored for four trials while $\mathcal{W}$ was favored for one trial. Note that we can also consider the hole to be attached to the gripper and the peg to the world, which would result in swapping the viewpoints between $\mathcal{TL}$ and $\mathcal{W}$.

The results for the orientation viewpoint are also plausible. For example, the significant choice for $\mathcal{TL}$ for the Revolute Joint follows from the variability in the demonstrations, with the joint in different orientations with respect to the world. The 3-D Contour Following task also exhibits a high ratio, attributed to the low variation in direction in $\mathcal{TL}$ for both force and translational velocity, whereas these directions vary a lot in $\mathcal{W}$. In contrast, the ratios for the Peg on Hole Alignment, 2-D Contour Following and Prismatic Joint indicate only slight preferences for $\mathcal{TL}$ or $\mathcal{W}$. This is in agreement with the limited variability of the orientation in the demonstrations.

Vectors of Interest. Overall the procedure yields plausible results for the vectors of interest. For motion, Revolute Joint and Peg on Hole Alignment have significant ratios, indicating predominant rotation according to Model 1. Conversely, 3-D Contour Following and Prismatic Joint exhibit high ratios, indicating predominant translation according to Model 2. For 2-D Contour Following, the ratio is closer to 1, indicating a smaller preference for translation

Table 3.4: Outcome of the proposed approach for selection of the task frame and comparison with task frames chosen by an expert.
The comparison with the expert for the Drawing task is visualized in Figure 3.14. For the rest of the tasks, please refer to the videos in the supplementary material for the visualization.

Task	**Viewpoint**		**Vectors of Interest**		**Derived Task Frame $\{tf\}$ vs. Expert's Task Frame $\{tf^*\}$**							
	Origin (Ratio)	**Orientation (Ratio)**	**Motion (Ratio)**	**Wrench (Ratio)**	**From Motion**		**From Wrench**		**Unweighted Avg.**		**Weighted Avg.**	
					[°]	[mm]	[°]	[mm]	[°]	[mm]	[°]	[mm]
Revolute Joint	$\mathcal{TL}$ ($1.3e^1$)	$\mathcal{TL}$ ($1.9e^1$)	$\boldsymbol{\omega}$ (6.9)	$\boldsymbol{f}$ (2.3)	2.3[a]	4.4[b]	9.9[a]	100.8[b]	**2.3**[a]	**4.4**[b]	**2.3**[a]	4.5[b]
Peg on Hole Alignment	$\mathcal{TL}$ (1.9)	$\mathcal{TL}$ (1.2)	$\boldsymbol{\omega}$ ($1.7e^1$)	$\boldsymbol{f}$ (3.5)	2.6[a]	15.9[b]	4.8[a]	42.0[b]	**2.0**[a]	**26.6**[b]	2.6[a]	27.5[b]
Bottle Opening	$\mathcal{TL}$ ($2.8e^1$)	$\mathcal{TL}$ (4.2)	$\boldsymbol{\omega}$ (2.2)	$\boldsymbol{m}$ (4.3)	2.7[a]	18.5[b]	12.1[a]	180.8[b]	**10.8**[a]	**26.0**[b]	12[a]	26.2[b]
Prismatic Joint	$\mathcal{TL}$ (1.2)	$\mathcal{TL}$ (1.3)	$\boldsymbol{v}$ ($1.3e^1$)	$\boldsymbol{f}$ (1.6)	1.3[a]	358.7[b]	37.1[a]	122.9[b]	**1.3**[a]	**166.5**[b]	1.3[a]	167.6[b]
Drawing	$\mathcal{TL}$ ($1.7e^3$)	$\mathcal{W}$ (9.5)	$\boldsymbol{v}$ (2.0)	$\boldsymbol{f}$ (2.6)	0.3[c]	498.8[d]	3.9[c]	8.9[d]	3.8[c]	**8.9**[d]	**3.7**[c]	**8.9**[d]
2-D Contour Following	$\mathcal{TL}$ ($6.8e^2$)	$\mathcal{W}$ (1.1)	$\boldsymbol{v}$ (1.6)	$\boldsymbol{f}$ (1.9)	4.1[c]	344.1[e]	5.9[c]	2.0[e]	8.5[c]	**2.0**[e]	**6.2**[c]	**2.0**[e]
3-D Contour Following	$\mathcal{TL}$ ($1.1e^3$)	$\mathcal{TL}$ ($1.8e^1$)	$\boldsymbol{v}$ ($6.1e^1$)	$\boldsymbol{f}$ ($5.1e^1$)	1.3[a]	46.6[d]	19.0[a]	33.6[d]	**2.5**[a]	**8.2**[d]	3.2[a]	**8.2**[d]

[a] Total angle between ${}^{[tf]}\boldsymbol{e}_x$ and ${}^{[tf^*]}\boldsymbol{e}_x$ [b] Magnitude of common normal between ${}^{[tf]}\boldsymbol{e}_x$ and ${}^{[tf^*]}\boldsymbol{e}_x$ [c] Total angle between ${}^{[tf]}\boldsymbol{e}_z$ and ${}^{[tf^*]}\boldsymbol{e}_z$
[d] $\left\| [p_x^* \; p_y^* \; p_z^*]^T - [p_x \; p_y \; p_z]^T \right\|$ [e] $\left\| [p_x^* \; p_y^*]^T - [p_x \; p_y]^T \right\|$

over rotation. This is attributed to the tool maintaining a constant orientation while translating in two DOFs, hence there is not a single predominant direction for translation in both $\mathcal{TL}$ and $\mathcal{W}$.

Concerning wrenches, Model 1 yields the most significant fit for 3-D Contour Following, because the force has quite a constant application point in $\mathcal{TL}$. Remarkably, Bottle Opening is the only task where Model 2 (constant moment) yields the best fit. Accordingly, the moment is selected as the wrench vector of interest.

Comparison with Expert's Choice. Some, but certainly not all, of the task frame candidates derived from only motion or only wrench data closely align with the expert task frame. For example, the motion candidate for Revolute Joint and the wrench candidate for 2-D Contour Following closely align with $\{tf^*\}$, both in origin and orientation. Also, it appears that motion candidates are often more accurate, which is attributed to the operator exhibiting more variability in the wrench space. All this underscores the importance of averaging the results of motion and wrench candidates, which may involve combining complementary information. The comparison with the expert for the Drawing task is visualized in Figure 3.14. In this application, the origin of the task frame is attached to the tool while the orientation of the task frame is attached to the world. For the rest of the tasks, please refer to the videos in the supplementary material for the visualization of the derived and expert task frames. Finally, our results indicate that the weighted average of motion and wrench orientation candidates does not yield substantial benefit compared to the unweighted average, at least for the experiments that we performed.

Influence of friction. Friction influences the orientation of the task frame derived from the wrench data. For instance, in the Drawing task, the task frame obtained from the wrench measurements deviates from the expert-defined frame by 3.9°. This deviation is primarily due to friction, although the direction of motion also plays a role. In contrast, the task frame derived from motion data deviates by only 0.3° from the expert frame. When the two candidate frames, based on motion and wrench, are combined according to their associated uncertainties, the resulting task frame deviates by 3.8°, which is closer to the wrench-based estimate. This outcome indicates that the final derived task frame inherently captures the influence of friction, i.e., while deviating from the expert task frame, the task frame description will compensate the effect of friction, which is not the case in the expert task description. In practice, if friction conditions change, the task frame can be updated by reprocessing data recorded during robot performance. This iterative adjustment allows the system to adapt to new frictional conditions, ensuring that the task frame remains representative of the current physical contact.

Conclusion. The experimental results reveal a favorable correspondence between the derived and expert task frames for all the studied applications. They also reveal that the derived task frame encodes the variability that is present in the demonstrations (or lack thereof). Although often less accurate, depending on the application, the wrench candidate proves valuable in determining the optimum origin and orientation in combination with the motion candidate. There is not a significant benefit in using a weighted average for determining the task frame orientation. However, we did not invest effort in tuning the four reference values, relying on initial guesses instead.

3.6.2 Control Performance

To assess the controller's performance, the tracking errors for all signals and all robot experiments are reported in Table 3.5 using the RMSE.

For every application, several scenarios were considered. Below, we present a brief overview of the results, ordered according to the scenarios introduced in Section 3.5.6.

Table 3.5: Robot control performance: tracking errors for all reference signals in all applications and scenarios.

Task	No.	ΔR^a [°]	Δp [mm]	$\Delta\omega$ [°/s]	Δv [mm/s]	Δf [N]	Δm [Nm]
Revolute Joint							
Nominal	1-1	0.3	8.3	0.1	4.8	4.0	0.5
Speed × 0.25	1-2	0.3	8.7	0.0	2.2	1.8	0.3
Peg on Hole Alignment							
Nominal	2-1	0.4	1.6	0.1	2.6	1.3	0.1
Wrench × 3	2-2	0.2	2.8	0.1	3.9	2.2	0.2
Peg Diameter × 0.7	2-3	0.2	3.9	0.0	2.9	1.3	0.0
Bottle Opening							
Nominal	3-1	1.1	1.6	3.8	2.2	7.4	0.3
Opener Length × 1.5	3-2	3.3	2.7	7.3	4.1	6.9	0.3
Prismatic Joint							
Nominal	4-1	0.1	3.8	0.0	1.5	1.9	0.3
Speed × 4.5	4-2	0.0	3.9	0.0	3.7	4.4	0.5
Rotated Task Frame	4-3	0.1	2.9	0.0	1.3	1.6	0.4
Drawing							
Nominal	5-1	0.1	5.5	1.3	3.1	2.1	0.2
Wrench × 0.5	5-2	0.1	5.4	1.2	3.0	2.1	0.1
Inclined Table	5-3	0.1	8.1	1.2	4.7	3.8	0.2
Inclined Table, Iterated	5-4	0.1	2.9	1.7	1.5	0.6	0.0
2-D Contour Following							
Nominal	6-1	0.1	4.4	0.6	2.6	1.1	0.2
No Pose Constraints[b]	6-2	14.2	80.8	2.4	4.4	1.8	0.2
3-D Contour Following							
Nominal	7-1	3.3	28.1	1.4	2.9	10.1	0.2
Nominal, Iterated	7-2	1.5	10.9	1.5	0.7	2.7	0.2
Changed Origin[b]	7-3	3.8	26.6	1.7	2.8	9.4	0.2

[a] The RMSE values for ΔR are calculated as $\|\boldsymbol{r}\|$, with $\boldsymbol{r}$ derived according to (3.4).

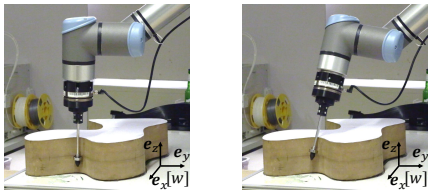
[b] Task performance was not successful.

Nominal Scenarios

The first six tasks were performed under conditions including task speed that closely resemble those of the demonstrations, see lines 1-1 to 6-1. They were all successful, with small values for the pose, twist and wrench tracking errors. For Bottle Opening, the force error Δf was higher than in the other tasks, almost doubled compared to the (small) average desired force. This may be attributed to the significant desired moment, averaging 1.1 Nm, and the expected variability of the required interaction wrench to open different bottles. The last task, 3-D Contour Following (7-1), was demonstrated with a translational velocity of approximately 45 mm/s and a contact force of 50 N, requiring a lot of skill and training by the operator to keep all five wheels in contact. For the performance by the robot to be successful, the translational velocity had to be reduced to 15 mm/s. The value of the force error Δf was large compared to the other tasks; however, it accounted for only 20% of the average value, 50 N, which was much

higher than in the other tasks. Additionally, the small orientation and position errors in all nominal[4] scenarios indicate that the calculation of the desired pose using the similarity transformation is analytically correct.

Role of pose constraints. Pose constraints are required to maintain a constant position or orientation of the tool in the absence of geometric constraints imposed by the environment. This is illustrated in the 2-D Contour Following application (6-2), where we omitted the pose constraints. This led to a failed task, because the orientation of the tool and its vertical position are not geometrically restricted, allowing them to deviate far from their desired values. This is depicted in Figure 3.10 and indicated by the large values of ΔR and Δp.



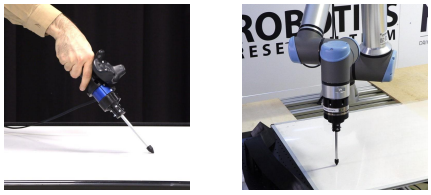
(a) (b)

Figure 3.10: 2-D Contour Following performed by the UR10e robot (snapshots of video): (a) nominal scenario with pose and wrench constraints; (b) nominal scenario without pose constraints, resulting in task failure.

Role of wrench constraints. Controlling the contact wrench allows to compensate for minor disturbances and small deviations from the demonstration environment. This is illustrated in a variation of the Drawing task by tilting the table 5° with respect to its demonstrated orientation without adjusting the tool's initial orientation (5-3). The inclined table causes the errors in translational direction, Δf, Δp and Δv, to increase with respect to the nominal scenario: to maintain contact, force feedback normal to the surface generates a compensating motion.

Independence of initial pose. This feature is illustrated in the nominal scenario of the Drawing task (5-1), see Figure 3.11: the operator demonstrated using a tilted pen, where the initial angle between the normal vector of the surface and the pen was approximately 30°, and approximately maintained a constant orientation during the task. In contrast, the robot started from an initial pose where the pen was approximately oriented along the normal of the surface, but was able to maintain a similar constant orientation with respect to this initial orientation. This is evidenced by the small value of ΔR.

[4]Or, for the case of 3-D Contour Following: Nominal, Iterated



(a) (b)

Figure 3.11: Difference in orientation of the tool in the Drawing task (snapshots of video): (a) tilted in the demonstrations; (b) almost vertical in the robot performance

.

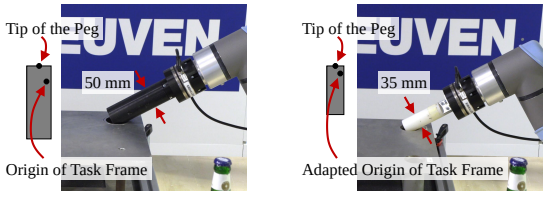
Variation of task speed. Task speed can easily be varied by changing the progress function, $\xi(t)$. An overall speed-up or slow-down is implemented by adjusting $\dot{\xi}$ with a constant factor. In the Revolute Joint application, the task was performed with 25% of the demonstrated velocity (1-2). The reduction in speed led to a more precise tracking of the desired values or maintained a comparable level of accuracy compared to the nominal case. This was according to expectation, as lower speeds allow the controller to track the desired values more accurately. The reduction was more pronounced for $\Delta\omega$, Δv, Δf and Δm. Conversely, to assess the controller's capabilities, the Prismatic Joint task was performed at its maximum speed of 180 $^{\text{mm}}/_{\text{s}}$, i.e. 4.5 times faster than the nominal case. The results (4-2) indicate a similar level of error for all variables compared to the nominal case, except for Δv and Δf which increased by a factor up to 2.5. As expected, the desired force could not be tracked as effectively as in the nominal case.

Variation of contact wrench. Similarly, the desired contact wrench can easily be rescaled. In the Peg on Hole Alignment application, the desired wrench was tripled (2-2). Accordingly, Δf and Δm almost double in comparison to the nominal case, but they remain small. Also Δp and Δv increase but remain small. Conversely, the desired wrench values were halved in the Drawing task, still yielding successful completion. The errors are reported in 5-2, consistently demonstrating equal or lower values for all errors compared to the nominal case.

Changed Geometry

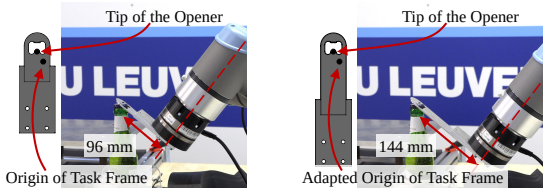
The original peg in the Peg on Hole Alignment task was replaced by a thinner and shorter one, while the hole diameter was reduced accordingly. This is illustrated in Figure 3.12. To adapt to this change, we intuitively scaled the

position of the task frame relative to a frame at the tip of the new peg in two directions, as illustrated in Figure 3.12 to the left of the pictures. The used scaling factor was $\frac{35\ \text{mm}}{50\ \text{mm}} = 0.7$. This adjustment enabled a successful task completion with no significant change in Δf and Δm compared to the nominal scenario, as reported in line 2-3.



(a) (b)

Figure 3.12: Peg on Hole Alignment performed by the UR10e robot (snapshots of video): (a) nominal scenario with peg diameter of 50 mm; (b) changed geometry with peg diameter of 35 mm.



(a) (b)

Figure 3.13: Bottle Opening performed by the UR10e (snapshots of video): (a) nominal scenario with same opener as in demonstrations; (b) changed geometry with a 50% longer opener.

Similarly, in the Bottle Opening application, the opener was replaced by another one which was 1.5 times longer. Again, we relied on intuition to relocate the task frame, as shown in Figure 3.13: the origin of the task frame remains unchanged with respect to to a frame attached to the lower inner tip of the opener. Due to the increased length of the opener, the compliance values $\hat{C}_f$ and $\hat{C}_m$ had to be multiplied by a factor of two. The task frame adjustment enabled a successful cap removal. However, as indicated in 3-2, compared to the nominal scenario we observed similar RMSE levels for Δf and Δm, but nearly doubled RMSEs for the remaining variables.

We could have followed a similar approach in the Drawing scenario with inclined table, i.e. adapt ${}^{tf}\boldsymbol{R}_{w}$ manually. Instead, illustrating another approach, we recorded the task performance by the robot with the inclined table (5-3), and reintroduced these data in the task frame derivation procedure to find the change in task frame orientation. Subsequently, we repeated the task performance with inclined table and with the new task frame (5-4). This resulted in smaller errors for almost all variables, even compared to the nominal case. This is not unexpected, because, using robot data instead of human demonstration data avoids calibration errors between demonstration and robot environment. This is an interesting finding. In a similar manner, we derived the task frame from robot data recorded during nominal 3-D Contour Following task and we repeated the task performance. The RMSEs decreased as shown in (7-2), especially for Δp and Δf. Note that calibration errors may also result from differences in compliance between the demonstration and robot performance setups, affecting the desired pose trajectory. This was particularly the case in the 3-D Contour Following application, as we introduced some compliance in the robot setup by putting the contour on a foam sheet. The smaller errors in the iterated robot performance allowed us to increase the task speed to 20-25 $^{\text{mm}}/_{\text{s}}$ while keeping all five wheels in contact.

Sensitivity to Task Frame Origin and Orientation

To confirm that the proposed controller is not dependent on the orientation of the task frame, we repeated the Prismatic Joint task with a 17.7° change in orientation for the task frame. As reported in 4-1 and 4-3, the error values remained at the same level, affirming the controller's robustness to variations in the task frame orientation. Furthermore, to illustrate the relevance of selecting an appropriate origin for the task frame, we changed the origin of the derived task frame by 23.4 mm in the 3-D Contour Following application. This resulted in a failed task, as some of the wheels lost contact. It is however challenging to confirm loss of contact at some wheels from the reported errors, as the robot kept on moving with some wheels out of contact, but close to the contour. However, the slightly higher values of ΔR and $\Delta\omega$ are an indication that the tool experienced more rotation than in the nominal case, turning away from the contour.

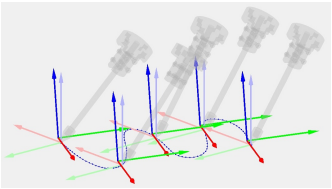


Figure 3.14: Expert task frame (pale) and derived task frame (bright) for the Drawing task according to the RGB convention. The origin of the task frame is attached to the tool while the orientation of the task frame is attached to the world. The derived task frames for the other applications are shown in videos in the supplementary material.

3.7 Discussion and Conclusion

Contributions. The main contribution of this chapter is an automatic procedure for deriving an optimal task frame for contact-rich robot tasks from demonstration data. The procedure is summarized in Figure 3.6 and fits in the red block of the overall workflow depicted in Figure 3.1b. The demonstration data may consist of human-demonstrated, simulation or robot data. The procedure is based on two principles that are hypothesized to underpin the control configuration targeted by an expert when choosing the task frame: (1) decoupled control of the directional and moment components of a screw, and (2) decoupled control of motion and interaction wrench. The procedure is rooted in screw theory and consists of a pipeline of algorithms. Besides existing algorithms such as ASIP and AVOF, the pipeline includes the following novelties: (1) an alternative application of ASIP (i.e. ASIP applied to a mean-subtracted screw), (2) a new algorithm to align orientation frames, (3) an adapted version of an algorithm to find the average of two orientations that can be applied to rotation matrices instead of quaternions, and (4) an extension of the concept of similarity transformation to pose matrices. The procedure is entirely probabilistic and does not involve any hyperparameters unless optional weighting is applied for averaging the orientation (but in our experiments we did not find evidence of significant added value for the latter). It also produces invariant results, regardless of the reference frames in which the data are expressed. The result is a gray-box model of the task, i.e. a set of parameters and reference signals that have unambiguous physical meaning and relations between them, and that are obtained in a data-driven way.

Main innovations with respect to the state of the art. The task frame derivation approach is fully automatic and fully data-driven. I.e., it is *generative*, as it

does not require predefined candidate reference frames or any input other than the demonstration data. Furthermore, the task frame is not constrained to be entirely fixed to the world or the tool as the origin and the orientation may be - independently - fixed to either world or tool. The approach systematically uses (1) both motion and wrench data and (2) complete screw data, i.e. both in translation and orientation, and force and moment. As a result, the approach is *generic*, as it applies to a large variety of contact-rich tasks.

Validation of the task frame derivation procedure. First, the task frame derivation procedure itself was validated using seven applications including surface following tasks (2-D and 3-D Contour Following, Drawing), manipulation of articulated objects (Revolute and Prismatic joints), a nonreciprocal task involving plastic deformation (Bottle Opening) and tasks involving multiple contact points (Peg on Hole Alignment, 3-D Contour Following). The tasks involved both activities of daily life and technical tasks. While the derived task frames showed a good agreement with the task frames assumed to be chosen by experts, it is important to note that, (1) depending on the type of application, the accuracy of all the components of the origin or orientation may not be equally relevant, (2) in some cases the derived frame may be more optimal than the expert's choice, which may be based on intuition or assumptions that are too simplified. Also, although the experiments were based on five demonstration trials for each application, the procedure can be applied to a single demonstration as well. Finally, our experiments confirmed the added value of combining motion and wrench data for deriving the optimal task frame.

Validation of the benefits of using a well-chosen and parameterized task frame model. A second set of validation experiments consisted of a robot performing the seven tasks based on the obtained task model (task frame parameters and reference signals for twist, wrench and pose). We designed a one-size-fits-all controller based on the combination of pose and wrench constraints. Hence, the controller was certainly not optimal for all applications – that was not the main focus of the chapter – but could be tuned quickly for each application with only three parameters. The experiments showed that, for each application, successful task completion and stable control could be achieved in the nominal scenario where the robot performs the task under the same conditions as observed during the demonstrations. Furthermore, thanks to the type of controller and the gray-box nature of the task model, we could demonstrate successful task performances in case of (1) a different starting pose, (2) a changed task speed, (3) a changed magnitude of the interaction wrench and (4) small changes to the geometric configuration. For larger changes in the geometric configuration of tool or environment, we demonstrated two solutions: (1) manual adaptation of the task frame parameters and (2) iteration of the derivation of the task frame parameters using the data of the task performed by the robot. The

latter option is interesting because such iteration eliminates calibration errors between the demonstration environment and the robot environment. Further investigation is required, but this option could potentially turn out to be a good standard practice after an application has been ported from demonstration to robot. Finally, it was shown that, due to its isotropic nature, the controller performed equally well regardless of the orientation of the task frame, while its performance deteriorated when the origin of the derived task frame was changed. The latter underscores the importance of selecting the optimal origin for the task frame.

Limitations and future work. Current limitations of the proposed method are the following. First, we started from data that had already been segmented. Future work could investigate whether changes in the outcomes of the task frame derivation procedure can be used to identify transitions between the contact-rich segments studied in this chapter and other types of segments: gross motions in free space, approach-to-contact motions, building up of an initial contact and moving out of contact.

Second, the task frame derivation procedure includes several binary decisions (e.g. about viewpoint or progress variable). The choice between the two options can sometimes be very arbitrary, just depending on noise and small human variations present in the data. However, we argue that, in such cases, both options yield a task model that will result in comparable robot task performance. Nevertheless, care has to be taken when the decision is used for other purposes, such as segmentation (see above). Therefore it is important to always consider also the statistical significance of the decision (i.e. the ratio of the determinants of the covariance matrices).

Third, we used reference signals obtained by only averaging the demonstration signals of the different trials after aligning them. We did not use more advanced statistics to deal with noise or with undesired human variations in the demonstration data. Also, we did not estimate the impedance observed in the human demonstrations and employ it in the robot controller, which is known to come at the cost of a more complicated experimental setup [4]. This was left for future work.

Fourth, we used a one-size-fits-all isotropic type of controller that does not exploit knowledge about contact dynamics or human impedance in different directions. Hence, although our work was inspired by hybrid control, we did not use a hybrid controller. While our approach is simple, it is expected that non-isotropic controllers can further improve the task performance. This is recommended for future work.

Fifth, we derived an *average* task frame for the *whole segment*, with a *fixed*

origin and a *fixed* orientation, both with respect to either world or tool. A future extension would be to derive an optimal task frame without these restrictions, i.e. an *instantaneous* task frame that is *completely free* to move. Such frame would be beneficial for controlling tasks with changing contact geometries, as in 2-D and 3-D Contour Following involving unknown objects or cutting. This could be achieved by considering the average task frame as proposed in this chapter, but over a shorter time window, or by switching to invariant motion and wrench trajectory descriptors as in [108].

Lastly, although the Bottle Opening application involved plastic deformation of the bottle cap, all our experiments involved rather stiff contacts. Hence, it would be worthwhile to investigate the suitability of our approach when manipulating more deformable objects.

Chapter 4

Estimating Force/Torque Sensor Offsets and Gravity Parameters Using Only Wrench Measurements to Facilitate Human Demonstration of Robot Manipulation Tasks in Contact*

Abstract

In human demonstration of manipulation tasks involving contact, a tool equipped with a force/torque (FT) sensor is typically used to capture contact wrenches

*This chapter is based on and extends preliminary research submitted as Mousavi Mohammadi, A. and Vochten, M., De Schutter, J., and Aertbeliën, E. Estimating force/torque sensor offsets and gravity parameters using only wrench measurements to facilitate human demonstration of robot manipulation tasks in contact, Accepted for publication in *IEEE International Conference on Automation Science and Engineering (CASE)*, (2025). Ali Mousavi Mohammadi was the main and corresponding author for this paper. He contributed to the methodology development, formal analysis, investigation, experimental validation, data curation, software development, and drafting of the original manuscript. Maxim Vochten contributed to the review and editing of the paper. Joris De Schutter contributed to the methodology development, formal analysis, and the review and editing of the paper. Erwin Aertbeliën provided supervision, contributed to the methodology development, formal analysis,

(i.e., forces and moments). For accurate measurements, the sensor must be properly calibrated to ensure that it only records the contact wrenches during tool-environment interactions. This estimation in our context refers to estimating the sensor offsets, as well as the gravity parameters, i.e. mass and center of mass (COM), of the rigid object mounted on the FT sensor. Proper estimation enables access to the true contact wrench, which facilitates construction of reliable task models from human demonstrations. We propose a method for estimating sensor offsets and gravity parameters using only wrench measurements. This is particularly beneficial in scenarios where orientation information is either unavailable or unreliable. By relying solely on wrench data, the method mitigates motion-related noise, eliminates the need for sensor orientation calibration, and remains computationally efficient, making it well-suited for real-time applications. The method's effectiveness is evaluated through both simulations and real-world experiments, using a baseline that requires both wrench and accurate orientation measurements, while our method uses only wrench data. In simulations, the bias and variance of both methods are analyzed. Although the baseline method generally yields more accurate estimates, the proposed method achieves comparable performance. In real-world experiments, both methods perform well, especially when the excitation exceeds 10°; however, the proposed method still shows consistently better performance across all experiments within the excitation range.

4.1 Introduction

In demonstration of manipulation tasks in contact, an operator holds a tool and interacts with the environment, as shown in Fig. 4.1. The true contact wrenches cannot be directly measured at the point of contact, necessitating the use of a force/torque (FT) sensor integrated somewhere between the interacting objects and the operator (or environment). However, since the FT sensor is not located at the contact point, disturbance wrenches arises due to the tool's weight, complicating the extraction of the true contact wrenches [59, 118, 50]. Moreover, strain-gauge-based FT sensors often exhibit offsets [59, 60] due to thermal, mechanical, or electrical factors, further distorting measurements. At high motion speeds and accelerations, dynamic wrenches can also introduce significant disturbances. However, most in-contact manipulation tasks are inherently slow, allowing the dynamic effects to be reasonably neglected [50]. Given these disturbance wrenches, compensating for gravitational effects and sensor offsets is essential to accurately obtain the true contact wrench. This

and the review and editing of the paper. The terminology and notation have been adjusted slightly for consistency throughout the thesis.

requires prior estimation, specifically estimating the gravity parameters, i.e. object's mass and center of mass (COM), and sensor offsets to ensure precise contact wrench measurements. Estimating these properties is critical not only for human demonstration scenarios but also for robotic applications demanding force control.

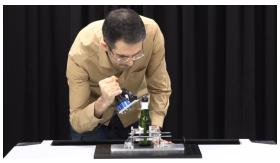


Figure 4.1: An operator demonstrates a manipulation task involving contact, where they use a bottle opener to remove the cap from a bottle.

Estimating gravity parameters typically involves a precise, tedious, and time-consuming process, particularly for gravity compensation, which requires manipulating the object in known orientations with respect to gravity.

Early studies, such as [13], utilized both pose and wrench data to estimate the inertia and gravity parameters of an object mounted on an FT sensor using least squares. In [109], the authors addressed sensor offsets and gravity parameter estimation using a least-squares approach; however, their method required the object to be positioned in specific orientations. Similarly, [59] applied the recursive instrumental variables method to estimate sensor offsets, inertia, and gravity parameters. Later, [20] extended least-squares-based estimation to include gravity parameters, sensor offsets, and the angle between the sensor and end-effector frames. In [99], a method was introduced to estimate the sensor's calibration matrix and offsets through two independent least-squares problems. Building upon this, [30] proposed a regularized least-squares approach to estimate the sensor's calibration matrix. Further developments include [111] and [117], which estimated sensor offsets, gravity parameters, and the installation angle bias of robot base (i.e. the vertical axis of base is not parallel to the direction of gravity vector) using a least-squares method. In [118], two approaches were introduced for estimating gravity parameters, sensor offsets, the rotation matrix between the sensor and the robot's end-effector frames, and the robot installation declination. The first approach required specific orientations, while the second relied on a least-squares formulation. More recently, [50] estimated sensor offsets, gravity parameters, and the rotation matrix between the robot's end-effector and sensor frames using the Gröbner basis method, refining these estimations via a non-linear least-squares problem. In parallel to these model-based methods, alternative approaches have emerged. For instance, [65] developed a back-propagation neural network-based method for gravity

compensation, using robot joint angles as inputs and FT sensor measurements as outputs. While this approach compensated for gravity, it did not directly estimate unknown parameters and required a large amount of data.

This chapter aims to develop a fast and accurate method for estimating the FT sensor offsets and gravity parameters of an unknown rigid object *without* dismounting it. The proposed approach relies *only* on wrench measurements from a single, arbitrary free-space manipulation trial, eliminating the need for orientation information, prior calibration, or additional sensing modalities such as vision or motion tracking. By removing these dependencies, the method simplifies the estimation process while maintaining high accuracy, making it particularly suitable for scenarios where motion data is either unavailable or unreliable, such as human demonstrations.

The key contribution of this work is the development of a novel estimation method that relies *only* on FT sensor data. To the best of the authors' knowledge, while estimation of gravity parameters, inertia, and FT sensor offsets using pose and wrench information is well-studied in robotics, estimation of gravity parameters and sensor offsets *without* knowledge of orientation remains unexplored, making this work the first to address it. The proposed method follows three steps: (1) estimating the *mass* and *force offsets* using a well-known sphere-fitting approach, which represents a novel application of this technique that can provide a closed-form solution, (2) estimating the *COM* through a novel screw theory-based formulation, also yielding a closed-form solution, and (3) determining *moment offsets*. By focusing exclusively on wrench measurements, the method mitigates motion-related noise and sensor orientation calibration. The method remains computationally efficient, making it particularly useful for real-time applications such as object sorting and task monitoring in unstructured environments.

The chapter is structured as follows. Section 4.2 provides an overview of the concept of wrench within the framework of screw theory, establishing the foundational principles relevant to this study. Section 4.3 presents the assumptions underlying our method and describes both a baseline method and the proposed method for estimating gravity parameters and sensor offsets. Sections 4.4 and 4.5 evaluate the performance of the baseline and proposed methods through simulation and real-world experiments, analyzing their accuracy and robustness. Section 4.6 summarizes the key findings, discusses their implications, and outlines potential directions for future research.

4.2 Preliminaries

The estimation method is rooted in screw theory. According to screw theory, the force and moment acting on a rigid body can be represented using a *screw wrench* [76, 42]:

$$\boldsymbol{w} = (\boldsymbol{f}^T \;\; \boldsymbol{m}^T)^T \in \mathbb{R}^{6\times 1}, \tag{4.1}$$

where $\boldsymbol{f}$ is the 3×1 force vector and $\boldsymbol{m}$ is the 3×1 moment vector seen at the origin of the reference frame. Each measurement sample from the FT sensor is represented as a screw wrench. Each screw wrench is associated with a screw axis, a line in space along which the force is oriented with a corresponding moment along the same line. An important property of the screw axis is that the moment along the screw axis has the smallest magnitude for the given screw wrench. For the remainder of the chapter, we will refer to the screw wrench simply as a *wrench*.

4.3 Estimating the Gravity Parameters and Sensor Offsets

4.3.1 Assumptions

A *rigid* object is mounted on the FT sensor. The wrench measurements $\boldsymbol{w} = (\boldsymbol{f}^T \;\; \boldsymbol{m}^T)^T$ consist of gravitational force $m\boldsymbol{g}$ and its resulting moment $\boldsymbol{p}_{com} \times m\boldsymbol{g}$, along with dynamic wrenches $\boldsymbol{w}_{dyn}$, contact wrenches $\boldsymbol{w}_{con}$, sensor offsets $\boldsymbol{w}_{off}$, and sensor noise $\boldsymbol{w}_{\eta}$.

$$\boldsymbol{f} = m\boldsymbol{g} + \boldsymbol{f}_{off} + \boldsymbol{f}_{dyn} + \boldsymbol{f}_{con} + \boldsymbol{f}_{\eta} \tag{4.2}$$

$$\boldsymbol{m} = \boldsymbol{p}_{com} \times m\boldsymbol{g} + \boldsymbol{m}_{off} + \boldsymbol{m}_{dyn} + \boldsymbol{m}_{con} + \boldsymbol{m}_{\eta} \tag{4.3}$$

Here, $\boldsymbol{p}_{com}$ represents the COM expressed in the FT sensor frame, m is the mass, and $\boldsymbol{g}$ denotes the gravity vector. However, we make the following assumptions for the estimation:

(a) Sensor noise, $\boldsymbol{f}_{\eta}$ and $\boldsymbol{m}_{\eta}$, can be approximated by zero-mean white noise.

(b) The object's motion during estimation is in *free space*, so contact forces and moments are zero: $\boldsymbol{f}_{con} = \boldsymbol{0}$ and $\boldsymbol{m}_{con} = \boldsymbol{0}$.

(c) The dynamic wrenches are *negligible*: $\boldsymbol{f}_{dyn} = \boldsymbol{0}$ and $\boldsymbol{m}_{dyn} = \boldsymbol{0}$. This assumption is reasonable for contact tasks, where speeds and accelerations

are typically low, but it is not satisfied in fast transfer motions, so this has to be taken into account when moving the object.

(d) Sensor offsets are assumed to remain *constant* throughout the estimation process and subsequent use of the sensor.

(e) The object mounted on the FT sensor is assumed to be a *rigid body*, with its gravity parameters (mass m and COM $\boldsymbol{p}_{com}$) treated as unknown and unmodeled.

With these assumptions, gravity is the only source of wrench applied to the object besides sensor offsets which are inherently present in the wrench measurements.

$$\boldsymbol{f} = m\boldsymbol{g} + \boldsymbol{f}_{off} + \boldsymbol{f}_{\eta} \tag{4.4}$$

$$\boldsymbol{m} = \boldsymbol{p}_{com} \times m\boldsymbol{g} + \boldsymbol{m}_{off} + \boldsymbol{m}_{\eta} \tag{4.5}$$

Fig. 4.2 can help to interpret these relations. When expressed in an arbitrary frame attached to the object, the wrench only depends on the orientation of the object: the gravity force changes in direction, while the gravity moment changes both in direction and magnitude.

To enable accurate estimation of the gravity parameters and sensor offsets, it is essential to provide sufficient excitation. Hence, it is advisable to generate arbitrary free-space motions with sufficient variation in orientation.

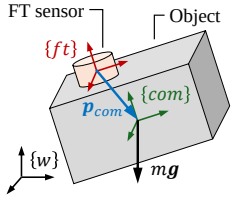


Figure 4.2: Visualization of the configuration of the FT sensor and the object mounted on it. The COM frame, denoted as $\{com\}$, is parallel to the FT sensor frame $\{ft\}$ and is located at the object's COM. The gravity force is always aligned with the vertical axis of the world frame $\{w\}$.

4.3.2 Baseline: Least-Squares Method

Under the given assumptions, each individual force $\boldsymbol{f}_i$ and moment $\boldsymbol{m}_i$ measurement i can be written using (4.4) and (4.5):

$$\boldsymbol{f}_i = m\boldsymbol{g}_i + \boldsymbol{f}_{off} + \boldsymbol{f}_\eta \tag{4.6}$$

$$\boldsymbol{m}_i = \boldsymbol{p}_{com} \times m\boldsymbol{g}_i + \boldsymbol{m}_{off} + \boldsymbol{m}_\eta. \tag{4.7}$$

The gravity vector $\boldsymbol{g}_i$ is the transformation of the gravity $\boldsymbol{g} = [0 \;\; 0 \;\; -g]^T$ from world frame to FT sensor frame using the inverse of a 3×3 rotation matrix of frame $\{ft\}$ with respect to frame $\{w\}$:

$$\boldsymbol{g}_i = {}^{ft}\boldsymbol{R}_{w,i}^{-1}\, \boldsymbol{g}. \tag{4.8}$$

After some algebraic manipulation, we obtain the following equations:

$$\boldsymbol{f}_i = m\boldsymbol{g}_i + \boldsymbol{f}_{off} + \boldsymbol{f}_\eta \tag{4.9}$$

$$\boldsymbol{m}_i = -[\boldsymbol{g}_i]_\times (m\boldsymbol{p}_{com}) + \boldsymbol{m}_{off} + \boldsymbol{m}_\eta. \tag{4.10}$$

where $[\boldsymbol{g}_i]_\times$ is the 3×3 skew-symmetric matrix corresponding to vector $\boldsymbol{g}_i$.[2]

Solving (4.9) and (4.10) for the unknown parameters can be formulated as a least-squares optimization problem:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\boldsymbol{w}_i - \boldsymbol{D}_i \boldsymbol{x}\|^2 \tag{4.11}$$

where $\boldsymbol{w}_i$ represents the wrench measurements and $\boldsymbol{D}_i$ is defined as:

$$\boldsymbol{D}_i = \begin{bmatrix} \boldsymbol{g}_i & \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0} & -[\boldsymbol{g}_i]_\times & \boldsymbol{0}_3 & \boldsymbol{I}_3 \end{bmatrix} \tag{4.12}$$

with $\boldsymbol{I}_3$ and $\boldsymbol{0}_3$ being the 3×3 identity and zero matrices, respectively. The vector of unknowns $\boldsymbol{x}$ consists of the mass, the product of mass and COM, and the sensor offsets, as follows:

$$\boldsymbol{x} = \begin{bmatrix} m & m\boldsymbol{p}_{com}^T & \boldsymbol{w}_{off}^T \end{bmatrix}^T. \tag{4.13}$$

[2]The skew-symmetric matrix of the vector $\boldsymbol{g}$:

$$[\boldsymbol{g}]_\times = \begin{bmatrix} 0 & -\boldsymbol{g}_z & \boldsymbol{g}_y \\ \boldsymbol{g}_z & 0 & -\boldsymbol{g}_x \\ -\boldsymbol{g}_y & \boldsymbol{g}_x & 0 \end{bmatrix}.$$

By stacking all the measurements, we obtain:

$$\boldsymbol{D} = \begin{bmatrix} \vdots \\ \boldsymbol{D}_i \\ \vdots \end{bmatrix} \quad \text{and} \quad \boldsymbol{W} = \begin{bmatrix} \vdots \\ \boldsymbol{w}_i \\ \vdots \end{bmatrix}. \tag{4.14}$$

The closed-form solution of the least-squares problem is given by the following expression:

$$\boldsymbol{x}^* = \boldsymbol{D}^\dagger \boldsymbol{W}, \tag{4.15}$$

where $\boldsymbol{D}^\dagger$ denotes the Moore-Penrose pseudo-inverse of $\boldsymbol{D}$. Note that the COM is not directly estimated, but since the mass is estimated independently, it can be determined accordingly.

This least-squares problem is similar to the formulation presented in [13], but with two key differences. First, [13] does not account for sensor offsets in the problem formulation, whereas we explicitly include them. Second, while [13] incorporates dynamic wrenches to estimate the object's inertia, we neglect them, assuming their contribution is very small at low motion speeds and accelerations. As explained in Section 4.3.1 (assumption (c)), one way to reduce these effects is by carefully selecting the motion used for estimation.

4.3.3 Proposed Three-Step Estimation Method

This subsection introduces a novel three-step method for estimating the gravity parameters and sensor offsets using only wrench measurements. The estimation process is structured as follows: (1) mass and force offsets, (2) COM, and (3) moment offsets. This procedure is visually represented in Fig. 4.3, providing an overview of the estimation method.

Step 1) Estimating the Mass and Force Offsets: This step estimates force offsets and mass using only force measurements. The approach is inspired by the problem of fitting a sphere to measured points that lie on the surface of the sphere. This can be analogously interpreted as estimating force offsets from measured forces. In this analogy, the sphere's center corresponds to the force offsets, while its radius represents the object's weight, given by $\boldsymbol{w} = m\boldsymbol{g}$. The 2D circle-fitting method from [32] is extended to 3D data points, i.e., force data. Given a set of force measurements $\{\boldsymbol{f}_i\}$, the force offsets $\boldsymbol{f}_{off}$ and mass m satisfy the following constraint:

$$\|\boldsymbol{f}_i - \boldsymbol{f}_{off}\|^2 = \|m\boldsymbol{g}\|^2. \tag{4.16}$$

Rearranging this equation, we obtain the following linear formulation:

$$\boldsymbol{f}_i^T \boldsymbol{f}_i = 2\boldsymbol{f}_i^T \boldsymbol{f}_{off} + (m^2 \|\boldsymbol{g}\|^2 - \boldsymbol{f}_{off}^T \boldsymbol{f}_{off}). \tag{4.17}$$

For a set of force measurements, we construct the following least-squares problem:

$$\boldsymbol{y}^* = \underset{\boldsymbol{y}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\boldsymbol{b}_i - \boldsymbol{E}_i \boldsymbol{y}\|^2, \tag{4.18}$$

where the unknown parameter vector $\boldsymbol{y}$ is defined as:

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{f}_{off}^T & m^2 \|\boldsymbol{g}\|^2 - \boldsymbol{f}_{off}^T \boldsymbol{f}_{off} \end{bmatrix}^T, \tag{4.19}$$

and the measurement matrices are:

$$\boldsymbol{E}_i = \begin{bmatrix} 2\boldsymbol{f}_i^T & 1 \end{bmatrix}, \; \boldsymbol{b}_i = \begin{bmatrix} \boldsymbol{f}_i^T \boldsymbol{f}_i \end{bmatrix}. \tag{4.20}$$

By stacking all the measurements, we obtain:

$$\boldsymbol{E} = \begin{bmatrix} \vdots \\ \boldsymbol{E}_i \\ \vdots \end{bmatrix} \quad \text{and} \quad \boldsymbol{B} = \begin{bmatrix} \vdots \\ \boldsymbol{b}_i \\ \vdots \end{bmatrix}. \tag{4.21}$$

The closed-form solution of the least-squares problem is given by the following expression:

$$\boldsymbol{y}^* = \boldsymbol{E}^\dagger \boldsymbol{B}. \tag{4.22}$$

Note that the mass does not appear directly in (4.19), but since the force offset is estimated independently, the mass, can be determined accordingly. Subtracting $\boldsymbol{f}_{off}$ from the force measurements yields force-offset-corrected wrench data.

Step 2) Estimating the COM: Given a set of wrench measurements influenced by gravity and sensor offsets, the moment at the COM should ideally remain *constant*. Since gravity does not contribute to the moment at the COM, these moment values correspond solely to the moment offsets. To account for these offsets, we subtract the mean wrench from the dataset before computing the COM.

The *Average Screw-axes Intersection Point* (ASIP) is defined as the approximate intersection point for a given set of N unordered screw axes associated with the wrenches [74]. The ASIP can estimate the point with the *minimal average moment* which corresponds to the definition of the COM where the moment is ideally zero. In [74], it is also demonstrated that subtracting the mean wrench from the dataset before applying ASIP yields the point with the *minimum variation in moment*. This refined approach is referred to as the *minimum-variation Average Screw-Axes Intersection Point* (min-var ASIP).

The min-var ASIP is determined by solving the following least-squares problem:

$$\boldsymbol{p}_{com} = \underset{\boldsymbol{p}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left\| \boldsymbol{f}_i \times \boldsymbol{p} + \boldsymbol{m}_i - \boldsymbol{m}_{off} \right\|^2 . \tag{4.23}$$

Noting that for the mean of the forces $\bar{\boldsymbol{f}}$ and the mean of the moments $\bar{\boldsymbol{m}}$ the following equation holds:

$$\frac{1}{N} \sum_{i=1}^{N} \left(\boldsymbol{f}_i \times \boldsymbol{p} + \boldsymbol{m}_i - \boldsymbol{m}_{off} \right) =$$

$$\bar{\boldsymbol{f}} \times \boldsymbol{p}_{com} + \bar{\boldsymbol{m}} - \boldsymbol{m}_{off} \approx \boldsymbol{0} \tag{4.24}$$

since the bias of the sensor noise is assumed to be modeled with m_{off}. This allows for the elimination of $\boldsymbol{m}_{off}$ by reformulating problem (4.23) in terms of zero-mean wrenches:

$$\boldsymbol{p}_{com} = \underset{\boldsymbol{p}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left\| \breve{\boldsymbol{f}}_i \times \boldsymbol{p} + \breve{\boldsymbol{m}}_i \right\|^2 . \tag{4.25}$$

where the zero-mean forces and moments are defined as $\breve{\boldsymbol{f}}_i = \boldsymbol{f}_i - \bar{\boldsymbol{f}}$ and $\breve{\boldsymbol{m}}_i = \boldsymbol{m}_i - \bar{\boldsymbol{m}}$. This ensures robustness against sensor offset variations. In case the mean-subtracted force vectors $\breve{\boldsymbol{f}}_i$ are all close to parallel, the solution of this optimization problem is not unique. By adding a regularization term $\epsilon \left\| \boldsymbol{p} - \boldsymbol{p}_0 \right\|^2$ to (4.25), the results can be biased towards a prior value $\boldsymbol{p}_0$ with a weight ϵ. When ϵ is kept small, the effect is negligible in the regular case. The closed-form solution of the regularized least-squares problem is given by the following expression:

$$\boldsymbol{p}_{com} = \left[\boldsymbol{A} + \epsilon \boldsymbol{I}_3 \right]^{-1} \left(\frac{1}{N} \sum_{i=1}^{N} (\breve{\boldsymbol{f}}_i \times \breve{\boldsymbol{m}}_i) + \epsilon \, \boldsymbol{p}_0 \right), \tag{4.26}$$

where $\boldsymbol{A} = \frac{1}{N} \sum_{i=1}^{N} [\breve{\boldsymbol{f}}_i]_\times [\breve{\boldsymbol{f}}_i]_\times^T$. Equation (4.26) can be incrementally updated when new wrench measurements arrive, making this method suitable for online applications such as monitoring.

The covariance matrix on the uncertainty of the COM estimate is given by [10]:

$$\boldsymbol{C}_{com} = \operatorname{var}(\boldsymbol{p}_{com}) = \hat{\sigma}^2 \left(\boldsymbol{A} + \epsilon \boldsymbol{I}_3 \right)^{-1}, \tag{4.27}$$

where the value of $\hat{\sigma}^2$ can be estimated from the residuals on the moment in (4.23):

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{N} \left\| \breve{\boldsymbol{f}}_i \times \boldsymbol{p}_{com} + \breve{\boldsymbol{m}}_i \right\|^2}{N(3N-3)}. \tag{4.28}$$

The main directions of variation can be identified via a singular value decomposition of the covariance matrix:

$$\boldsymbol{C}_{com} = \boldsymbol{U}_{com}\ \boldsymbol{\Sigma}_{com}\ \boldsymbol{U}_{com}^{T}, \tag{4.29}$$

with $\boldsymbol{U}_{com}$ the matrix of singular vectors and $\boldsymbol{\Sigma}_{com}$ a diagonal matrix containing the singular values $\boldsymbol{C}_{com}$: $\sigma_1^2 \geq \sigma_2^2 \geq \sigma_3^2$. The first column of $\boldsymbol{U}_{com}$ indicates the direction with the most uncertainty.

Step 3) Estimating the Moment Offsets: Using the estimated COM $\boldsymbol{p}_{com}$, the moment offsets can be determined according to (4.24) as follows.

$$\boldsymbol{m}_{off} \approx \bar{\boldsymbol{f}} \times \boldsymbol{p}_{com} + \bar{\boldsymbol{m}} \tag{4.30}$$

Subtracting $\boldsymbol{m}_{off}$ from the moment measurements yields offset-corrected wrench data.

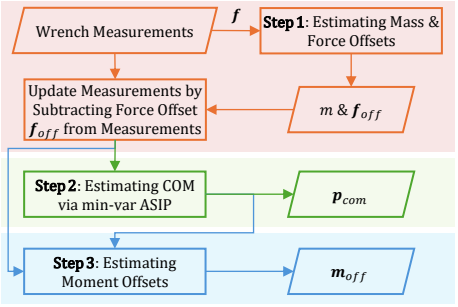


Figure 4.3: Overview of the proposed method. This includes three main steps: (1) estimation of mass and force offsets, (2) estimation of COM, and (3) estimation of moment offsets.

4.4 Simulation Results

This section analyzes performance of the baseline and the proposed methods by conducting a *Monte Carlo simulation* to verify the estimates of the unknowns and the computed variance on the estimates. Additionally, the biases introduced in different scenarios are examined, as well as the performance with low and high excitation. For each case, 1000 trials with synthetic wrench data are simulated and probabilistically analyzed.

To analyze the methods, the gravity wrench is simulated. It involves applying a pure force at the COM that remains vertical in the world frame and has a

constant magnitude. This wrench is then transformed from the COM to the FT sensor frame which introduces a moment component due to the moment arm. Noise is applied to the force and moment measurements according to a normal distribution with a standard deviation of 3.1×10^{-3} N for force and a standard deviation of 8.8×10^{-5} Nm for moment, where these values are extracted from real data recorded in a static condition. To simulate unknown offsets on the FT sensor, a force offset of $(3.6\ 2.8\ 0.8)^T$ N is applied to the force measurements, while a moment offset of $(0.6\ 0.3\ 0.6)^T$ Nm is applied to the moment measurements. The ground truth object mass is 1.5 kg while the ground truth position vector from the FT sensor to the COM is $(0.1\ \ -0.2\ 0.4)^T$ m. The total number of wrench measurements is taken as 20 for each trial.

At the FT sensor frame, an arbitrary free-space rotation is applied to introduce excitation in the wrench measurements. The rotations are specified with respect to a fixed world reference frame using three Euler angles in XYZ format. Two ranges of excitation of the Euler angles are considered (see Figure 4.4): a low range of excitation (α_1, β_1, and γ_1) and a high range of excitation (α_2, β_2, and γ_2). Note that with an alternative orientation representation such as roll-pitch-yaw angles, only two angles (roll and pitch) would be needed, since the yaw is along the vertical axis, i.e., in the direction of gravity, and therefore has no effect. Nevertheless, we use XYZ Euler angles since they can be directly related to the orientation of the robot's wrist axes.

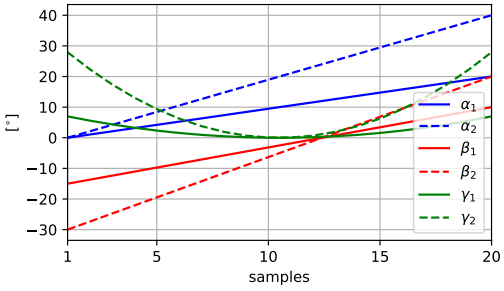


Figure 4.4: Range of excitation shown by Euler angles in XYZ format with angles α, β, and γ, respectively. Lower excitation ranges are shown in solid lines while the higher excitation ranges are shown in dashed lines.

Ten scenarios are designed to investigate the effects of sensor noise, sensor offset, and excitation range on the estimation methods. The simulation results are detailed in Table 4.1. The estimator bias is quantified by the Euclidean distance between the ground truth values and the average of the estimated unknowns

over all simulations. The variance of the estimator is quantified by averaging the standard deviation over all simulations.

To validate the estimation, the root mean square (RMS) values of force and moment, after compensating for weight and sensor offsets, are computed as follows:

$$\Delta f = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\Delta \boldsymbol{f}_i)^2}\ , \quad \Delta m = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\Delta \boldsymbol{m}_i)^2} \tag{4.31}$$

where $\Delta \boldsymbol{f}_i$ and $\Delta \boldsymbol{m}_i$ are defined as follows according to (4.4) and (4.5):

$$\Delta \boldsymbol{f}_i = \boldsymbol{f}_i - m\boldsymbol{g}_i - \boldsymbol{f}_{off,i} \tag{4.32}$$

$$\Delta \boldsymbol{m}_i = \boldsymbol{m}_i - \boldsymbol{p}_{com} \times m\boldsymbol{g}_i - \boldsymbol{m}_{off,i}. \tag{4.33}$$

Ideally, these values should be zero, hence RMS values serve as key validation criteria for assessing the accuracy of the estimation performed by both methods.

We analyze the results for each scenario individually.

Simulation 1, Effect of Offset and Noiseless Force and Moment: In the presence of offset but no noise in force and moment, both methods estimate the unknowns perfectly. This shows that both methods are robust to the values of offset.

Simulation 2, Effect of Offset and Noisy Moment: In the presence of offset and noise only in moment, the results slightly degrade compared to simulation 1. This indicates that both methods handle noise in the moment effectively.

Simulation 3, Effect of Offset and Noisy Force: In the presence of offset and noise only on force, the results significantly deteriorate compared to simulation 1. This means both methods perform well with noise in the moment but exhibits more bias with noise in the force. This is more pronounced for the proposed method. However, the RMS values of the proposed method remain comparable to those of the baseline, suggesting that it still provides a reliable estimation.

Simulation 4, Effect of Offset and Noisy Force and Moment: In the presence of offset and noise in force and moment, the estimation results are poor for both methods, with the proposed method performing worse than the baseline. However, the RMS values remain similar to those observed in the previous simulation, indicating that overall error levels are not significantly higher despite the reduced estimation quality.

Simulation 5, Effect of Offset-Free and Noisy Force and Moment: In the presence of noise in force and moment and without offset, the results are the same as simulation 4, meaning both methods are robust to offset values.

Table 4.1: Monte Carlo simulation for estimating the gravity parameters and sensor offsets using the baseline and proposed methods.

No.	Excitation Range	Offset	Noise		Method	Estimated Mass m [kg]		Estimated COM $\boldsymbol{p}_{com}$ [mm]		Estimated Force Offsets $\boldsymbol{f}_{off}$ [N]		Estimated Moment Offsets $\boldsymbol{m}_{off}$ [Nm]		Force RMS [N]		Moment RMS [Nm]	
		$\boldsymbol{f}$ & $\boldsymbol{m}$	$\boldsymbol{f}$	$\boldsymbol{m}$		Bias[a]	Standard Deviation[b]	Bias[a]	Standard Deviation[b]	Bias[a]	Standard Deviation[b]	Bias[a]	Standard Deviation[b]	Bias[a]	Standard Deviation[b]	Bias[a]	Standard Deviation[b]
1	Low	✓	✗	✗	Baseline	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
					Proposed	0.000	0.000	0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2		✓	✗	✓	Baseline	0.000	0.000	0.004	0.064	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000
					Proposed	0.000	0.000	0.009	0.064	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000
3		✓	✓	✗	Baseline	0.000	0.000	0.003	0.117	0.000	0.004	0.000	0.000	0.003	0.000	0.002	0.000
					Proposed	0.001	0.008	0.171	0.976	0.009	0.084	0.005	0.046	0.003	0.000	0.002	0.000
4		✓	✓	✓	Baseline	0.000	0.000	0.006	0.117	0.000	0.004	0.000	0.001	0.003	0.000	0.002	0.000
					Proposed	0.001	0.008	0.175	0.977	0.009	0.084	0.005	0.046	0.003	0.000	0.002	0.000
5		✗	✓	✓	Baseline	0.000	0.000	0.006	0.117	0.000	0.004	0.000	0.001	0.003	0.000	0.002	0.000
					Proposed	0.001	0.008	0.175	0.977	0.009	0.084	0.005	0.046	0.003	0.000	0.002	0.000
6	High	✓	✗	✗	Baseline	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
					Proposed	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
7		✓	✗	✓	Baseline	0.000	0.000	0.001	0.014	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
					Proposed	0.000	0.000	0.001	0.014	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
8		✓	✓	✗	Baseline	0.000	0.000	0.002	0.059	0.000	0.002	0.000	0.000	0.003	0.000	0.002	0.000
					Proposed	0.000	0.002	0.016	0.193	0.001	0.016	0.000	0.007	0.003	0.000	0.002	0.000
9		✓	✓	✓	Baseline	0.000	0.000	0.003	0.06	0.000	0.002	0.000	0.000	0.003	0.000	0.002	0.000
					Proposed	0.000	0.002	0.016	0.194	0.001	0.016	0.000	0.007	0.003	0.000	0.002	0.000
10		✗	✓	✓	Baseline	0.000	0.000	0.003	0.06	0.000	0.002	0.000	0.000	0.003	0.000	0.002	0.000
					Proposed	0.000	0.002	0.016	0.194	0.001	0.016	0.000	0.007	0.003	0.000	0.002	0.000
11		✓	✓	✓	Baseline	0.015	0.000	71.95	0.059	1.232	0.002	0.743	0.000	0.280	0.001	0.114	0.000
					Proposed	0.000	0.002	0.016	0.194	0.001	0.016	0.000	0.007	0.003	0.000	0.002	0.000

[a]For each case, the estimator bias represents the Euclidean distance between the ground truth and the average estimation over all trials.
[b]The standard deviation along the direction of the largest estimation uncertainty is reported, after averaging it over all trials.

Simulation 6, Effect of Offset and Noiseless Force and Moment with Higher Excitation Range: This scenario is identical to simulation 1 but with a higher excitation range. Both methods achieve perfect estimation, highlighting the benefit of increased excitation.

Simulation 7, Effect of Offset and Noisy Moment with Higher Excitation Range: With the same scenario as simulation 2 but with a higher excitation range, both methods show reduced bias and standard deviation in the estimates.

Simulation 8, Effect of Offset and Noisy Force with Higher Excitation Range: With the same scenario as simulation 3 but with a higher excitation range, both methods result in reduced bias and standard deviation in the estimates.

Simulation 9, Effect of Offset and Noisy Force and Moment with Higher Excitation Range: This scenario is the same as simulation 4 but with a higher excitation range. The estimation improves for both methods, showing lower bias and standard deviation in the estimates.

Simulation 10, Effect of Offset-Free and Noisy Force and Moment with Higher Excitation Range: Using the same scenario as simulation 5 but with a higher excitation range results in lower bias and standard deviation in the estimation.

Simulation 11, Effect of Offset and Noisy Force and Moment with Higher Excitation Range and Orientation Error: In the presence of offset and noise in the force and moment, a calibration error is introduced into the data. Specifically, a 5° orientation error around the X-axis in the sensor frame is assumed to be unaccounted for. Compared to simulation 9, the results show a significant degradation in the performance of the baseline method, particularly in the estimation of the COM, force offsets, and moment offsets. This orientation error also results in slightly higher values of RMS compared to simulation 9. In contrast, the proposed method remains unaffected, as it relies solely on wrench measurements. This scenario highlights the robustness of the proposed method and its ability to outperform the baseline method, even in the presence of small calibration errors.

This analysis suggests that accurately estimating the unknowns is challenging when both offset and noise affect the force and moment measurements. In such cases, the baseline method provides the best performance, as seen in simulations 4 and 9, due to its lower estimator bias and variance. Notably, the proposed method shows substantial improvement as the excitation range increases. Despite this, the methods yield comparable performance in terms of RMS values across most simulations, indicating that the proposed method can achieve results similar to the baseline while relying solely on wrench data. The next step is to evaluate both methods using real-world measurements.

4.5 Experimental Validation and Results

4.5.1 Experimental Setup

Two experimental setups are used, each consisting of an FT sensor and an object mounted on it

Robot Setup: The first setup, shown in Fig. 4.5a, consists of the manipulated object fixed to the robot's end-effector with a Schunk FT sensor mounted in between the robot and the object. In this configuration, the motion of the FT sensor frame $\{ft\}$ is directly available from the robot joint encoders and the robot's kinematic model, while the wrench measurements are recorded by the FT sensor.

Human Demonstration Setup: The second setup, depicted in Fig. 4.5b, is designed for human demonstration and features the manipulated object attached to a handheld tool with an FT sensor mounted in between. On the handheld tool, an HTC Vive motion tracker can be attached to measure its motion. Here, the motion of the tracker frame $\{tr\}$ is recorded with respect to the world frame, and the wrench measurements are captured by the FT sensor. To determine the motion of the FT frame $\{ft\}$, the transformation matrix between the tracker and the FT sensor frames must be calibrated.

Table 4.2 provides detailed specifications of the FT sensors used in both setups. While the object remains the same in both cases, differences in the mounting flanges and the number of screws used for attachment lead to variations in the estimated gravity parameters.

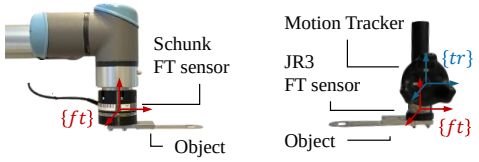


Figure 4.5: Setups used for the experimental validation. (a) The object and the Schunk FT sensor are mounted on the end-effector of the UR10e robot arm. (b) The object and the JR3 FT sensor as well as a motion tracker are mounted on a demonstration tool to be held by a human operator.

Table 4.2: Specifications of the force/torque sensors

6-Axis Force/Torque sensor	**Nominal Accuracy**	**Measurement Range**		
		Force [N]		**Torque** [Nm]
		f_x, f_y	f_z	τ_x, τ_y, τ_z
JR3 50M31A3	1%	±100	±200	±5
Schunk FTN-AXIA80	< 2%	±150	±470	±8

4.5.2 Experimental Methodology

Four experiments are conducted to analyze the proposed method for estimating the gravity parameters and sensor offsets. The data are collected at 200 Hz, and a first-order low-pass filter with a cutoff frequency of 1 Hz is applied to attenuate high-frequency noise and smooth the wrench signals. The first and the second experiments involve a robot setup, while the third and fourth are designed for a human demonstration, shown in Figs. 4.6 and 4.7, respectively. The data from each experiment is resampled to 1000 samples to ensure that the sample size does not affect the estimation results.



(a) (b) (c) (d) (e)

Figure 4.6: Experimental setup for analyzing the estimation of mass, COM, and sensor offsets with the robot. The object is rotated with a specified angle (here: ±70°) around two perpendicular axes, the X and Y axes of the FT sensor frame.



(a) (b) (c) (d) (e)

Figure 4.7: Experimental setup for analyzing the estimation of gravity parameters and sensor offsets within the human demonstration context. The operator is asked to rotate the object similarly to the motion of the robot's end-effector.

In the first experiment, the robot sequentially rotates the object around the $+X$, $-X$, $+Y$, and $-Y$ axes of the FT sensor frame, each by 70°, as shown in Fig. 4.6. The rotations around the $\pm X$ and $\pm Y$ axes of the FT sensor follow a trapezoidal velocity profile, where the maximum velocity is 0.4 $^{\text{rad}}/_{\text{s}}$ and the maximum acceleration is 1 $^{\text{rad}}/_{\text{s}^2}$. To analyze the effect of the size of excitation range, specific segments corresponding to maximum rotations of 5°, 10°, 20°, 40°, and 70° are cut, labeled as experiments 1-1 to 1-5 in Table 4.3. This approach enables a direct comparison of sensor offsets estimation results at different excitation ranges. If these rotations were treated as separate experiments, each would include different sensor offsets. By analyzing sensor offsets within a *single* experiment, we aim to assess how excitation range influences sensor offsets estimation accuracy and to compare these effects across various excitation ranges. Additionally, this approach allows for analyzing the impact of excitation range on gravity parameters estimation. In experiment 1-6, the number of samples is reduced to 100 to examine its effect on the results. In experiment 1-7, raw wrench measurements are used instead of those filtered by the low-pass filter. In experiment 1-8, artificial sensor offsets are introduced to account for potential variations and to evaluate their effect on the estimation of unknowns. Specifically, an artificial offset $\boldsymbol{w}_{art} = [3 \;\; 2 \;\; -5 \;\; 1 \;\; 2 \;\; 1.5]^T$ is added to the wrench measurements. In experiment 1-9, an artificial orientation error around the X-axis of the FT sensor (5°) is introduced to assess its effect on the baseline method. In experiment 1-10, the min-var ASIP used in Step 2 is replaced with the basic version of ASIP [74] without average wrench subtraction to evaluate the impact of this change.

In the second experiment, labeled as experiment 2 in Table 4.3, the same procedure is repeated but at a *lower velocity*. The maximum velocity is reduced from 0.4 $^{\text{rad}}/_{\text{s}}$ to 0.05 $^{\text{rad}}/_{\text{s}}$ and the maximum acceleration is reduced from 1 $^{\text{rad}}/_{\text{s}^2}$ to 0.1 $^{\text{rad}}/_{\text{s}^2}$. The goal is to analyze the effect of dynamic wrenches on the estimation accuracy.

In the third experiment, labeled as experiment 3 in Table 4.3, a *human operator* rotates the object freely in space, as shown in Fig. 4.7. The operator is instructed to perform rotations of approximately 70° to ensure comparability with the robot experiments. Unlike the robot's controlled rotations, these motions introduce natural variability, allowing us to test the method in a human demonstration scenario.

Finally, in the fourth experiment, labeled as experiment 4 in Table 4.3, a *heavier* object is mounted on the JR3 FT sensor, and the operator performs free-space rotations similar to the previous experiment. This test assesses whether the method remains effective for objects with greater mass.

The baseline and proposed methods of Section 4.3 are then applied to estimate the unknowns, including gravity parameters and sensor offsets. It is important to note that the proposed method *only* relies on wrench measurements and does not require orientation measurements, whereas the baseline method utilizes *both* orientation and wrench measurements.

Although the proposed method quantifies uncertainty using the standard deviations σ_1, σ_2, and σ_3 along the three principal directions, we report only σ_1, the standard deviation along the direction of the largest estimation uncertainty, in Table 4.3.

In all experiments, RMS values are calculated *within* the excitation range, while for experiments 1-1 to 1-4, they are also evaluated at *full* range to assess the validity of the estimation derived from a smaller excitation range. It is important to note that, in practice, providing ground truth values for comparison is often challenging. Therefore, small RMS values are generally interpreted as an indication of good estimation quality.

4.5.3 Experimental Results

The results of the experiments are presented in Table 4.3. We analyze the results for each experiment individually.

Experiments 1-1 to 1-5, Effect of Excitation Range: The COM estimation remains highly accurate even under low excitation ranges in both methods. Additionally, the uncertainty in COM estimation decreases as the excitation range increases. Both methods achieve a valid estimation within the excitation range, as indicated by the RMS values of force and moment, which remain close to zero, as shown by Δf and Δm under *RMS within Range* in Table 4.3. However, increasing the excitation range improves estimation accuracy beyond the excitation range, particularly for Δf, as evidenced by the significant

Table 4.3: Estimating the gravity parameters and sensor offsets using the baseline and proposed methods.

Setup	**No.**	**Excitation Range** [°]	**Method**	**Estimated Mass** [kg]	**Estimated COM** $\boldsymbol{p}_{com}$ [mm]			**Uncertainty** [mm]	**Estimated Force Offsets** $\boldsymbol{f}_{off}$ [N]			**Estimated Moment Offsets** $\boldsymbol{m}_{off}$ [Nm]			**RMS within Range** [N]	[Nm]	**RMS at Full Range**[a] [N]	[Nm]
				m	$p_{com,x}$	$p_{com,y}$	$p_{com,z}$	σ_1	$f_{off,x}$	$f_{off,y}$	$f_{off,z}$	$m_{off,x}$	$m_{off,y}$	$m_{off,z}$	Δf	Δm	Δf	Δm
Robot	1-1	5	Baseline	0.501	17.309	18.277	11.042	-	0.064	-0.120	-10.133	-0.029	-0.019	0.006	0.013	0.001	0.288	0.002
			Proposed	0.275	17.210	18.159	10.221	0.038	0.060	-0.102	-7.895	0.012	-0.058	0.006	0.012	0.000	0.613	0.002
	1-2	10	Baseline	0.438	17.472	18.054	10.567	-	0.058	-0.113	-9.516	-0.017	-0.029	0.006	0.015	0.000	0.103	0.002
			Proposed	0.354	17.319	18.101	10.320	0.026	0.086	-0.092	-8.681	-0.002	-0.044	0.006	0.011	0.000	0.213	0.002
	1-3	20	Baseline	0.416	17.416	18.090	10.461	-	0.057	-0.112	-9.307	-0.013	-0.033	0.006	0.017	0.000	0.034	0.002
			Proposed	0.396	17.382	18.048	10.478	0.015	0.093	-0.089	-9.094	-0.010	-0.037	0.006	0.011	0.000	0.050	0.002
	1-4	40	Baseline	0.413	17.421	18.011	10.462	-	0.055	-0.114	-9.283	-0.012	-0.034	0.006	0.021	0.001	0.026	0.002
			Proposed	0.405	17.374	17.983	10.584	0.010	0.090	-0.097	-9.181	-0.011	-0.035	0.006	0.012	0.001	0.026	0.002
	1-5	70	Baseline	0.411	17.290	17.833	10.447	-	0.046	-0.123	-9.273	-0.011	-0.035	0.006	0.028	0.001	-	-
			Proposed	0.408	17.270	17.841	10.608	0.012	0.071	-0.120	-9.219	-0.010	-0.036	0.005	0.020	0.001	-	-
	1-6	70	Baseline	0.411	17.288	17.830	10.452	-	0.046	-0.124	-9.273	-0.011	-0.035	0.006	0.027	0.001	-	-
			Proposed	0.408	17.267	17.838	10.608	0.037	0.071	-0.120	-9.225	-0.010	-0.036	0.005	0.019	0.001	-	-
	1-7	70	Baseline	0.415	17.300	17.874	10.415	-	0.046	-0.122	-9.299	-0.011	-0.035	0.006	0.059	0.002	-	-
			Proposed	0.408	17.264	17.841	10.609	0.012	0.071	-0.118	-9.200	-0.009	-0.036	0.005	0.054	0.001	-	-
	1-8	70	Baseline	0.411	17.290	17.833	10.447	-	3.046	1.877	-14.273	0.989	1.965	1.506	0.028	0.001	-	-
			Proposed	0.408	17.270	17.841	10.608	0.012	3.071	1.880	-14.219	0.990	1.964	1.505	0.020	0.001	-	-
	1-9	70	Baseline	0.410	17.275	18.011	9.705	-	0.046	-0.379	-9.254	-0.008	-0.035	0.001	0.122	0.003	-	-
			Proposed	0.408	17.270	17.841	10.608	0.012	0.071	-0.120	-9.219	-0.010	-0.036	0.005	0.020	0.001	-	-
	1-10	70	Proposed	0.408	25.392	15.601	10.704	0.157	0.071	-0.120	-9.219	-0.003	-0.012	0.005	0.020	0.018	-	-
	2	70	Baseline	0.417	17.268	17.872	10.358	-	0.024	-0.127	-9.430	-0.012	-0.035	0.006	0.036	0.001	-	-
			Proposed	0.410	17.176	17.862	10.586	0.012	0.060	-0.116	-9.338	-0.010	-0.036	0.005	0.026	0.001	-	-
Human	3	-	Baseline	0.260	28.581	28.996	-27.901	-	0.105	0.104	2.351	0.072	-0.073	-0.001	0.077	0.004	-	-
			Proposed	0.240	28.080	29.490	-28.430	0.036	0.102	0.110	2.114	0.067	-0.065	-0.000	0.077	0.004	-	-
	4	-	Baseline	2.510	0.283	0.120	-137.303	-	-0.827	0.973	10.775	0.148	0.004	-0.002	0.669	0.149	-	-
			Proposed	2.450	-0.437	0.535	-136.852	0.284	0.207	0.318	10.073	0.066	-0.124	-0.003	0.471	0.148	-	-

[a] The full range is 70°. Thus, the estimation derived from data within smaller excitation ranges is evaluated using new data obtained from the full excitation range.

reduction in its values under *RMS at Full Range*. Interestingly, Δm remains unchanged, which is attributed to the accurate COM estimation achieved even with low excitation ranges. Experiment 1-5 can be considered a ground truth reference, as it has the highest excitation range. The RMS values of force and moment in a new experiment for the unloaded, dismounted Schunk FT sensor at rest, are $\Delta f = 0.011$ N and $\Delta m = 0.000$ Nm, respectively. The RMS values for experiment 1-5 ($\Delta f = 0.020$ N and $\Delta m = 0.001$ Nm) are of a similar order and remain very small, confirming that the estimation is acceptable.

Experiment 1-6, Effect of Sample Size: Compared to experiment 1-5, the results remain consistent regardless of the number of samples used. The only notable difference is a higher COM uncertainty in experiment 1-6, which is expected since the number of samples appears in the denominator of (4.28), so using fewer samples naturally increases the uncertainty. These results show that both methods work reliably, even when the sample size changes. It's also worth mentioning that wrench measurements are usually correlated over time. By downsampling the data, we significantly reduce this correlation. However, the estimation results stay almost the same, which further supports the robustness of the methods.

Experiment 1-7, Effect of Using Raw Wrench Measurements: Compared to experiment 1-5, using raw wrench measurements instead of filtered ones does not produce any significant difference in the estimation results. This indicates that, although applying a low-pass filter may introduce some delay in the wrench signals and slight misalignment between motion and wrench data, it does not noticeably affect the performance of the baseline method.

Experiment 1-8, Invariance to Sensor Offsets: Compared to experiment 1-5, the estimated sensor offsets shift by exactly the added artificial offsets $\boldsymbol{w}_{art} = [3 \;\; 2 \;\; -5 \;\; 1 \;\; 2 \;\; 1.5]^T$ while the mass and COM remain unchanged, confirming that variations in offsets do not affect the estimation of mass and COM.

Experiment 1-9, Effect of Artificial Orientation Error in the FT Sensor Frame: When an artificial orientation error is introduced in the FT sensor frame, the proposed method remains unaffected as it does not rely on orientation measurements, and thus, its results remain unchanged. However, the baseline method produces new estimation results with larger values for Δf and Δm compared to 1-5, highlighting a major drawback: even small errors in sensor orientation calibration can significantly impact the accuracy of the estimation of the unknown parameters.

Experiment 1-10, Effect of Replacing min-var ASIP with ASIP: Replacing min-var ASIP with ASIP leads to significantly different estimates for the COM and

moment offsets compared to experiment 1-5. Accordingly, the higher values of Δm indicate that ASIP provides a less accurate model in the presence of moment offsets. In addition, the uncertainty in COM estimation increases significantly when using ASIP (0.157 mm) compared to min-var ASIP (0.012 mm in 1-5). Overall, these results demonstrate the benefits of using min-var ASIP instead of using ASIP.

Experiment 2, Effect of Slower Motion: Reducing the velocity does not improve the estimation of unknowns. The RMS values of force Δf and moment Δm remain at the same order of magnitude as in experiment 1-5, indicating that estimation accuracy is not affected by the motion speed and acceleration used in the experiments. However, if the speed and acceleration increase significantly, the effects of dynamic wrenches might not be negligible anymore.

Experiment 3, Human Demonstration: Both methods provide comparable results. To validate the estimation, we compare the RMS values of force and moment ($\Delta f = 0.077$ N and $\Delta m = 0.004$ Nm) with the RMS values of the unloaded, dismounted JR3 FT sensor at rest ($\Delta f = 0.045$ N and $\Delta m = 0.001$ Nm). Since these values are of a similar order and remain small, we conclude that the estimation is valid.

Experiment 4, Human demonstration with a Heavy Object: Again, the estimation results from both methods are comparable. The only significant difference from experiment 3 is an increase in COM uncertainty, which may be attributed to acceleration and dynamic effects, as human motion is not inherently smooth.

Overall, both methods perform well with an excitation range higher than 10°, yielding small RMS values, but the proposed method slightly outperforms the baseline when the estimation results are used *within* the excitation range. This is particularly evident in the *RMS within Range* values, where the proposed method consistently achieves lower or equal values across all experiments. This may be attributed to the pose calibration errors of the motion and FT sensors. When used *outside* the excitation range (under *RMS at Full Range*), the baseline method only provides a better estimation model if the excitation range of the experiment remains limited (i.e. less than 40° in our experiments).

4.6 Conclusions

The main contribution of this study is the development of a novel, fast, and closed-form method which uses *only* wrench measurements to estimate the FT sensor offsets and gravity parameters (mass and COM). Reliance only on wrench measurements eliminates the necessity for especially designed motions,

orientation calibration between motion and FT sensors, and minimizing the influence of potential inaccuracies in motion data. The simulation and real-world results show that even small calibration errors can significantly degrade the performance of the baseline method, whereas the proposed method remains robust. This highlights that, although recording motion data is relatively inexpensive and straightforward, achieving accurate calibration of the FT and motion sensors remains a challenging task. By relying solely on wrench measurements, the proposed method effectively mitigates this issue. Based on real-world experiments, both the proposed method and the baseline perform well when the excitation range exceeds 10°. However, across all experiments, the proposed method consistently outperformed the baseline method within the excitation range, despite relying solely on wrench measurements, whereas the baseline method incorporates both orientation and wrench data. For future work, the method's ability to provide uncertainty estimates in different directions could be leveraged to guide and evaluate motion excitation concurrently. Additionally, COM estimation could be applied to online tasks like object sorting and task monitoring, such as tracking liquid containers (e.g., jugs) to monitor progress during actions like pouring.

Chapter 5

Relations Between Invariant Descriptors and Moving Frames for Relative Motion and Interaction Wrench Between Two Rigid Bodies

Abstract

In Chapter 2 invariant descriptors were already introduced for vector and screw trajectories. In Section 2.5.5, it was shown that these invariants can be calculated with respect to either a frame fixed to the world or one attached to the tool. This chapter aims to derive the mathematical relations between the moving frames and their corresponding invariants defined based on data expressed in: world (object 1) and tool (object 2). These mathematical relations represent the main novelty of this chapter, even though they do not have a direct practical application.

5.1 Introduction

Considering the relative motion and interaction wrench between two rigid bodies, each represented by their respective frames, this chapter aims to establish the relation between invariant descriptors and their associated moving frames. The analysis focuses on two object frames, denoted as $\{o_1\}$ and $\{o_2\}$. Depending on which object is chosen as the reference, the invariants for motion and wrench, along with their associated moving frames, will differ.

Although this chapter focuses on the relations between vector/screw invariants calculated with respect to two objects, the approach can be extended to include scenarios involving multiple objects. In addition to object frames, invariants can also be calculated within a reference frame (task frame) constructed using the approach proposed in Chapter 3. This implies that the origin and orientation of the task frame can be rigidly attached to different objects and interpreted as a frame associated with a virtual rigid body. Consequently, the relative motion and interaction wrench can still be modeled using invariant descriptors, while the relations between invariants across this additional frame can be generalized based on the derivations presented for two objects.

The first objective of this chapter is to mathematically define the relations between the vector invariants and the corresponding Frenet-Serret FS frames for general vector trajectories with respect to two objects, i.e. in two viewpoints. Specifically, it addresses the relative motion between two general FS frames, $[mf_1]$ and $[mf_2]$, in terms of orientation and vector invariants. These general vector trajectories may represent rotational motion, translational motion, force, or moment trajectories.

The second objective is to mathematically define the relations between the screw invariants and the corresponding Instantaneous Screw Axis ISA frames for general screw trajectories expressed in two viewpoints. This involves describing the relative motion between two general ISA frames, $\{mf_1\}$ and $\{mf_2\}$, in terms of pose and screw invariants. Such general screw trajectories may include motion or wrench trajectories.

The relations between invariants and their associated moving frames derived from two viewpoints are discussed in Sections 5.2 and 5.3, respectively. These relations are mathematically derived in detail for vector and screw trajectories in Appendices E and F, respectively. Appendix G validates the mathematical formulations using synthetic and real experimental data. Finally, Section 5.4 concludes the chapter.

5.2 Relations between Moving Frames

The goal of this section is to derive relations between moving frames. A FS frame, being an orientation frame, is represented using square brackets, $[mf]$, while an ISA frame, as a pose frame, is represented using curly brackets, $\{mf\}$.[1] This distinction implies that the analysis of FS frames is conducted at the *orientation* level, whereas the analysis of ISA frames is performed at the *pose* level.

5.2.1 FS Frames

The relation between the orientations of the FS frames at *orientation* level derived from a general vector $\boldsymbol{c}$ represented in two viewpoints is explained in detail in Appendix E. In the first viewpoint, the vector is represented as ${}^{o_2}_{o_1}\boldsymbol{c}^{o_2}_{o_1}$, whereas in the second viewpoint, the vector is represented as ${}^{o_1}_{o_2}\boldsymbol{c}^{o_2}_{o_2}$.[2] The associated moving frames for these viewpoints are denoted by $[mf_1]$ and $[mf_2]$, respectively. These moving frames are defined relative to their corresponding reference frames as ${}^{mf_1}_{o_1}\widetilde{\boldsymbol{R}}_{o_1}$ and ${}^{mf_2}_{o_2}\widetilde{\boldsymbol{R}}_{o_2}$. This vector $\boldsymbol{c}$ can be replaced by $\boldsymbol{\omega}$, $\boldsymbol{v}$, $\boldsymbol{f}$, and $\boldsymbol{m}$. Table 5.1 provides a brief summary of the notation and names of the FS frames.

Table 5.1: Name of different FS frames according to the type of the vector $\boldsymbol{c}$.

Vector	**Viewpoint 1,** $\mathscr{o}_1$			**Viewpoint 2,** $\mathscr{o}_2$		
	Vector Notation	**Orientation Frame**	**Rotation Matrix**	**Vector Notation**	**Orientation Frame**	**Rotation Matrix**
General Vector $\boldsymbol{c}$	${}^{o_2}_{o_1}\boldsymbol{c}^{o_2}_{o_1}$	$[mf_1]$	${}^{mf_1}_{o_1}\widetilde{\boldsymbol{R}}_{o_1}$	${}^{o_1}_{o_2}\boldsymbol{c}^{o_2}_{o_2}$	$[mf_2]$	${}^{mf_2}_{o_2}\widetilde{\boldsymbol{R}}_{o_2}$
Rotational Motion $\boldsymbol{\omega}$	${}^{o_2}_{o_1}\boldsymbol{\omega}_{o_1}$	$[rot_1]$	${}^{rot_1}_{o_1}\widetilde{\boldsymbol{R}}_{o_1}$	${}^{o_1}_{o_2}\boldsymbol{\omega}_{o_2}$	$[rot_2]$	${}^{rot_2}_{o_2}\widetilde{\boldsymbol{R}}_{o_2}$
Translational Motion $\boldsymbol{v}$	${}^{o_2}_{o_1}\boldsymbol{v}^{o_2}_{o_1}$	$[tra_1]$	${}^{tra_1}_{o_1}\widetilde{\boldsymbol{R}}_{o_1}$	${}^{o_1}_{o_2}\boldsymbol{v}^{o_2}_{o_2}$	$[tra_2]$	${}^{tra_2}_{o_2}\widetilde{\boldsymbol{R}}_{o_2}$
Force $\boldsymbol{f}$	${}^{o_2}_{o_1}\boldsymbol{f}_{o_1}$	$[for_1]$	${}^{for_1}_{o_1}\widetilde{\boldsymbol{R}}_{o_1}$	${}^{o_1}_{o_2}\boldsymbol{f}_{o_2}$	$[for_2]$	${}^{for_2}_{o_2}\widetilde{\boldsymbol{R}}_{o_2}$
Moment $\boldsymbol{m}$	${}^{o_2}_{o_1}\boldsymbol{m}^{o_2}_{o_1}$	$[mom_1]$	${}^{mom_1}_{o_1}\widetilde{\boldsymbol{R}}_{o_1}$	${}^{o_1}_{o_2}\boldsymbol{m}^{o_2}_{o_2}$	$[mom_2]$	${}^{mom_2}_{o_2}\widetilde{\boldsymbol{R}}_{o_2}$

The relations between different frames are shown by a scene graph at *orientation* level in Figure 5.1. This graph includes rotational motion and a general case that could be replaced by translational motion, force, or moment. The rotational part, shown in pale green in Figure 5.1, is a special case and should be considered alongside the analysis of translational motion, force, or moment. As will be

[1] This is consistent with the List of Symbols.

[2] As explained in the List of Symbols, ${}^{b}_{c}\boldsymbol{c}^{d}_{a}$ is a 3×1 general vector representing a quantity associated with rigid body *"b"* with respect to rigid body *"a"* at reference point d and expressed in reference orientation frame $[c]$.

shown at the end of this subsection, in this special case, the relation between the moving frames of rotation $[rot_1]$ and $[rot_2]$ simplifies significantly.

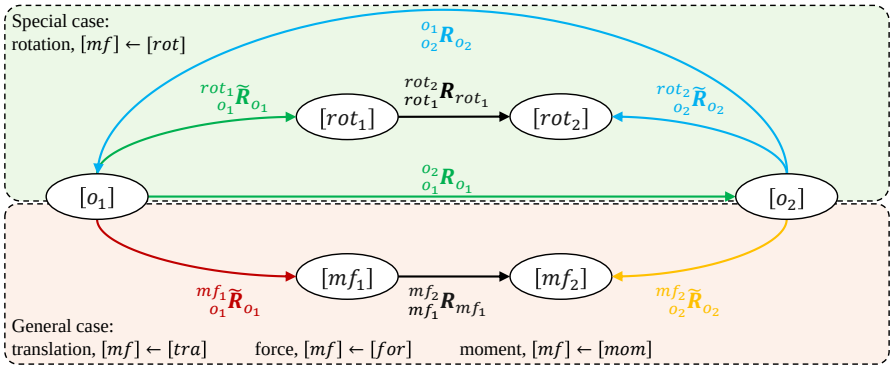


Figure 5.1: Scene graph at *orientation* level for frames, including object frames and FS frames. Frames in two viewpoints explained for rotation as the special case (in pale green) and a general case which could be translation, force, or moment (in pale red).

Appendix E provides the relation between the orientation of FS frames as described by (E.20):

$$ {}^{mf_2}_{mf_1}\boldsymbol{R}_{mf_1} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -\mathrm{c}\phi_c & \mathrm{s}\phi_c \\ 0 & \mathrm{s}\phi_c & \mathrm{c}\phi_c \end{pmatrix}, \tag{E.20} $$

where ϕ_c describes a rotation around the tangent axis of the FS frame which can be found according to (E.19). This relation is shown in Figure 5.2a. As explained in Appendix E, in case of $\boldsymbol{c} = \boldsymbol{\omega}$, ϕ_ω is zero. Then, this special case results in a simpler form of (E.20) as follows:

$$ {}^{rot_2}_{rot_1}\boldsymbol{R}_{rot_1} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{E.21} $$

This relation is shown in Figure 5.2b where the two FS frames have a fixed relative orientation, specifically, a 180° flip.

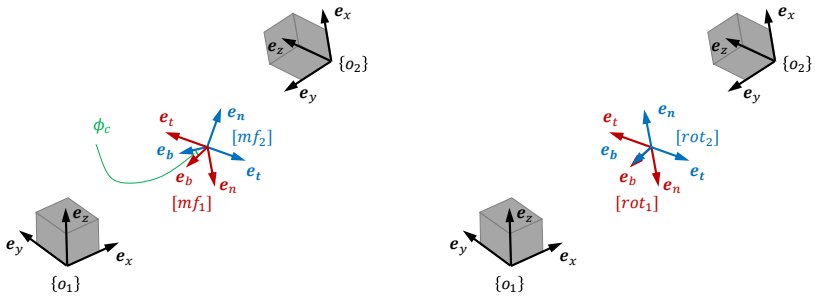


Figure 5.2: FS frames calculated in two viewpoints for: (a) a general case based on ${}^{o_2}_{o_1}\boldsymbol{c}^{o_2}_{o_1}$ and ${}^{o_1}_{o_2}\boldsymbol{c}^{o_2}_{o_2}$ expressed in $\{o_1\}$, (b) the special case of rotation based on ${}^{o_2}_{o_1}\boldsymbol{\omega}_{o_1}$ and ${}^{o_1}_{o_2}\boldsymbol{\omega}_{o_2}$ expressed in $\{o_1\}$.

5.2.2 ISA Frames

Considering $\boldsymbol{\jmath}$ as a general screw, we can derive the relation between the poses of the ISA frames obtained from two viewpoints. The relative orientation of the ISA frames follows the same formulation as that developed for the FS frames in Appendix E, since the orientation of an ISA frame is determined based on the directional component of a screw $\boldsymbol{\jmath} = (\boldsymbol{a}^T \quad \boldsymbol{b}^T)^T$, i.e. $\boldsymbol{a}$. This directional component is itself a vector, as discussed in more detail in Section 2.4.1. The relation between the positions of the ISA frames at *pose* level is derived in Appendix F for a general screw $\boldsymbol{\jmath}$. In the first viewpoint, the screw is represented as ${}^{o_2}_{o_1}\boldsymbol{\jmath}^{o_1}_{o_1}$, whereas in the second viewpoint, it is represented as ${}^{o_1}_{o_2}\boldsymbol{\jmath}^{o_2}_{o_2}$.[3] The associated moving frames for these viewpoints are denoted by $\{mf_1\}$ and $\{mf_2\}$, respectively. These moving frames are defined relative to their corresponding reference frames as ${}^{mf_1}_{o_1}\widetilde{\boldsymbol{T}}_{o_1}$ and ${}^{mf_2}_{o_2}\widetilde{\boldsymbol{T}}_{o_2}$. The screw $\boldsymbol{\jmath}$ can be either a twist $\boldsymbol{t}$ or a wrench $\boldsymbol{w}$ as shown in Table 5.2.

[3]As explained in the List of Symbols, ${}^{b}_{c}\boldsymbol{\jmath}^{d}_{a}$ is a 6×1 general screw representing a quantity associated with rigid body *"b"* with respect to rigid body *"a"* at reference point d and expressed in reference orientation frame $[c]$. This screw consists of two 3×1 vectors known as directional component ${}^{b}_{c}\boldsymbol{a}_{a}$ and moment component ${}^{b}_{c}\boldsymbol{b}^{d}_{a}$.

Table 5.2: Name of different ISA frames according to the type of the screw $\boldsymbol{\jmath}$.

Screw	Viewpoint 1, $\mathcal{O}_1$			Viewpoint 2, $\mathcal{O}_2$		
	Screw Notation	Pose Frame	Transformation Matrix	Screw Notation	Pose Frame	Transformation Matrix
General Screw $\boldsymbol{\jmath}$	${}^{o_2}_{o_1}\boldsymbol{\jmath}^{o_1}_{o_1}$	$\{mf_1\}$	${}^{mf_1}_{o_1}\widetilde{\boldsymbol{T}}_{o_1}$	${}^{o_1}_{o_2}\boldsymbol{\jmath}^{o_2}_{o_2}$	$\{mf_2\}$	${}^{mf_2}_{o_2}\widetilde{\boldsymbol{T}}_{o_2}$
Twist $\boldsymbol{t}$	${}^{o_2}_{o_1}\boldsymbol{t}^{o_1}_{o_1}$	$\{isam_1\}$	${}^{isam_1}_{o_1}\widetilde{\boldsymbol{T}}_{o_1}$	${}^{o_1}_{o_2}\boldsymbol{t}^{o_2}_{o_2}$	$\{isam_2\}$	${}^{isam_2}_{o_2}\widetilde{\boldsymbol{T}}_{o_2}$
Wrench $\boldsymbol{w}$	${}^{o_2}_{o_1}\boldsymbol{w}^{o_1}_{o_1}$	$\{isaw_1\}$	${}^{isaw_1}_{o_1}\widetilde{\boldsymbol{T}}_{o_1}$	${}^{o_1}_{o_2}\boldsymbol{w}^{o_2}_{o_2}$	$\{isaw_2\}$	${}^{isaw_2}_{o_2}\widetilde{\boldsymbol{T}}_{o_2}$

The relations between different frames are shown for a general screw and a twist by a scene graph at *pose* level in Figure 5.3. In this context, twist represents a special case, as it will be shown at the end of this subsection that the relation between the ISA frames simplifies significantly in this case. In the figure, the special case of twist is depicted in pale green.

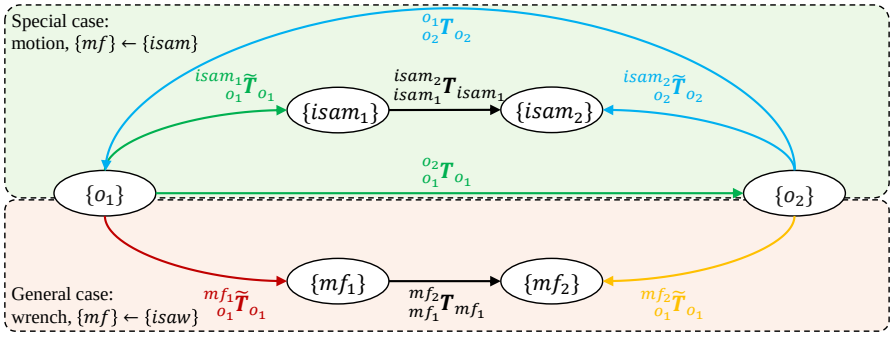


Figure 5.3: Scene graph at *pose* level for frames, including object frames and ISA frames. Frames in two viewpoints explained for motion as the special case (in pale green) and a general case that could represent a wrench (in pale red).

The relation between the poses of the ISA frames can be described in terms of their orientations and positions by (E.20) and (F.5), respectively.

$$ {}^{mf_2}_{mf_1}\boldsymbol{R}_{mf_1} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -\mathrm{c}\phi_c & \mathrm{s}\phi_c \\ 0 & \mathrm{s}\phi_c & \mathrm{c}\phi_c \end{pmatrix} \tag{E.20} $$

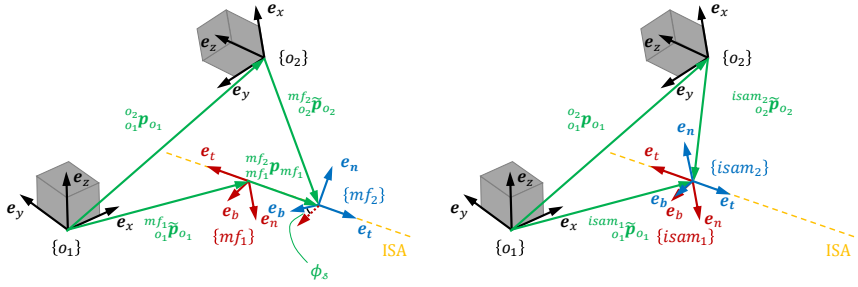
$$ {}^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}_{o_1} + {}^{mf_1}_{o_1}\widetilde{\boldsymbol{R}}_{o_1}\ {}^{mf_2}_{mf_1}\boldsymbol{p}_{mf_1} = {}^{o_2}_{o_1}\boldsymbol{p}_{o_1} + {}^{o_2}_{o_1}\boldsymbol{R}_{o_1}\ {}^{mf_2}_{o_2}\widetilde{\boldsymbol{p}}_{o_2} \tag{F.5} $$

Note that ϕ_c in (E.20) should be replaced by $\phi_\jmath$ in the context of screws. Using ${}^{mf_2}_{mf_1}\boldsymbol{R}_{mf_1}$ and ${}^{mf_2}_{mf_1}\boldsymbol{p}_{mf_1}$, we can also present the relation between the poses of

the ISA frames as a transformation matrix as follows:

$$ {}^{mf_2}_{mf_1}\boldsymbol{T}_{mf_1} = \begin{pmatrix} -1 & 0 & 0 & p_t \\ 0 & -\mathrm{c}\phi_\jmath & \mathrm{s}\phi_\jmath & 0 \\ 0 & \mathrm{s}\phi_\jmath & \mathrm{c}\phi_\jmath & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{F.24} $$

The relation between the ISA frames is visualized in Figure 5.4a where the frames are displaced along and rotated about the tangent axis. In Appendices E and F, it is shown that $\phi_\jmath$ and p_t can be found by (E.19) and (F.6), respectively.



(a) (b)

Figure 5.4: ISA frames calculated in two viewpoints for: (a) a general case based on ${}^{o_2}_{o_1}\boldsymbol{\jmath}^{o_1}_{o_1}$ and ${}^{o_1}_{o_2}\boldsymbol{\jmath}^{o_2}_{o_2}$ expressed in $\{o_1\}$ (b) the special case of motion based on ${}^{o_2}_{o_1}\boldsymbol{t}^{o_1}_{o_1}$ and ${}^{o_1}_{o_2}\boldsymbol{t}^{o_2}_{o_2}$ expressed in $\{o_1\}$.

In the special case of $\boldsymbol{\jmath} = \boldsymbol{t}$, the transformation matrix (F.24) simplifies as follows:

$$ {}^{isam_2}_{isam_1}\boldsymbol{T}_{isam_1} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{F.38} $$

This is explained in detail in Appendix F, and illustrated in Figure 5.4b, where the two ISA frames have a fixed relative orientation (a 180° flip), while their origins coincide.

5.3 Relations between Invariants

Vector invariants and screw invariants are inherently rotational velocity and twist, respectively. Therefore, to establish the relation between the invariants

derived from two viewpoints, the analysis must be performed at the *rotational velocity* level for vector invariants and at the *twist* level for screw invariants.

5.3.1 Vector Invariants

Table 5.3 presents vector invariants in two viewpoints, following the notation introduced in Chapter 2. Unlike Table 2.3, where the notation is simplified for clarity, the notation in this table is intentionally kept complete for comprehensive reference. At each viewpoint, there are object invariant $\widetilde{\boldsymbol{c}}$ and moving frame invariant $\boldsymbol{i}$. The object invariant relates to the object since c represents the magnitude of the vector. The moving frame invariant relates to the motion of the moving frame since it models the first-order kinematics of the moving frame.

Table 5.3: Vector invariants with complete notation in two viewpoints.

Vector	Viewpoint	Object Invariant	Moving Frame Invariant
Vector $\boldsymbol{c}$	$\mathcal{O}_1$	${}^{o_2}_{mf_1}\widetilde{\boldsymbol{c}}_{o_1} = (c^{mf_1} \;\; 0 \;\; 0)^T$	${}^{mf_1}_{mf_1}\boldsymbol{i}_{o_1} = (\omega_\tau^{mf_1} \;\; 0 \;\; \omega_\kappa^{mf_1})^T$
	$\mathcal{O}_2$	${}^{o_1}_{mf_2}\widetilde{\boldsymbol{c}}_{o_2} = (c^{mf_2} \;\; 0 \;\; 0)^T$	${}^{mf_2}_{mf_2}\boldsymbol{i}_{o_2} = (\omega_\tau^{mf_2} \;\; 0 \;\; \omega_\kappa^{mf_2})^T$
Rotational Motion $\boldsymbol{\omega}$	$\mathcal{O}_1$	${}^{o_2}_{rot_1}\widetilde{\boldsymbol{\omega}}_{o_1} = (\omega^{rot_1} \;\; 0 \;\; 0)^T$	${}^{rot_1}_{rot_1}\boldsymbol{i}_{o_1} = (\omega_\tau^{rot_1} \;\; 0 \;\; \omega_\kappa^{rot_1})^T$
	$\mathcal{O}_2$	${}^{o_1}_{rot_2}\widetilde{\boldsymbol{\omega}}_{o_2} = (\omega^{rot_2} \;\; 0 \;\; 0)^T$	${}^{rot_2}_{rot_2}\boldsymbol{i}_{o_2} = (\omega_\tau^{rot_2} \;\; 0 \;\; \omega_\kappa^{rot_2})^T$
Translational Motion $\boldsymbol{v}$	$\mathcal{O}_1$	${}^{o_2}_{tra_1}\widetilde{\boldsymbol{v}}_{o_1} = (v^{tra_1} \;\; 0 \;\; 0)^T$	${}^{tra_1}_{tra_1}\boldsymbol{i}_{o_1} = (\omega_\tau^{tra_1} \;\; 0 \;\; \omega_\kappa^{tra_1})^T$
	$\mathcal{O}_2$	${}^{o_1}_{tra_2}\widetilde{\boldsymbol{v}}_{o_2} = (v^{tra_2} \;\; 0 \;\; 0)^T$	${}^{tra_2}_{tra_2}\boldsymbol{i}_{o_2} = (\omega_\tau^{tra_2} \;\; 0 \;\; \omega_\kappa^{tra_2})^T$
Force $\boldsymbol{f}$	$\mathcal{O}_1$	${}^{o_2}_{for_1}\widetilde{\boldsymbol{f}}_{o_1} = (f^{for_1} \;\; 0 \;\; 0)^T$	${}^{for_1}_{for_1}\boldsymbol{i}_{o_1} = (\omega_\tau^{for_1} \;\; 0 \;\; \omega_\kappa^{for_1})^T$
	$\mathcal{O}_2$	${}^{o_1}_{for_2}\widetilde{\boldsymbol{f}}_{o_2} = (f^{for_2} \;\; 0 \;\; 0)^T$	${}^{for_2}_{for_2}\boldsymbol{i}_{o_2} = (\omega_\tau^{for_2} \;\; 0 \;\; \omega_\kappa^{for_2})^T$
Moment $\boldsymbol{m}$	$\mathcal{O}_1$	${}^{o_2}_{mom_1}\widetilde{\boldsymbol{m}}_{o_1} = (m^{mom_1} \;\; 0 \;\; 0)^T$	${}^{mom_1}_{mom_1}\boldsymbol{i}_{o_1} = (\omega_\tau^{mom_1} \;\; 0 \;\; \omega_\kappa^{mom_1})^T$
	$\mathcal{O}_2$	${}^{o_1}_{mom_2}\widetilde{\boldsymbol{m}}_{o_2} = (m^{mom_2} \;\; 0 \;\; 0)^T$	${}^{mom_2}_{mom_2}\boldsymbol{i}_{o_2} = (\omega_\tau^{mom_2} \;\; 0 \;\; \omega_\kappa^{mom_2})^T$

We assume that the vector invariants in two viewpoints are available for a general vector $\boldsymbol{c}$ as in the first line of Table 5.3. The object and moving frame invariants in the first viewpoint are given as follows:

$$ {}^{o_2}_{mf_1}\widetilde{\boldsymbol{c}}_{o_1} = \begin{pmatrix} c^{mf_1} & 0 & 0 \end{pmatrix}^T \tag{5.1} $$

$$ {}^{mf_1}_{mf_1}\boldsymbol{i}_{o_1} = \begin{pmatrix} \omega_\tau^{mf_1} & 0 & \omega_\kappa^{mf_1} \end{pmatrix}^T, \tag{5.2} $$

while the object and moving frame invariants in the second viewpoint are as follows:

$$ {}^{o_1}_{mf_2}\widetilde{\boldsymbol{c}}_{o_2} = \begin{pmatrix} c^{mf_2} & 0 & 0 \end{pmatrix}^T \tag{5.3} $$

$$ {}^{mf_2}_{mf_2}\boldsymbol{i}_{o_2} = \begin{pmatrix} \omega_\tau^{mf_2} & 0 & \omega_\kappa^{mf_2} \end{pmatrix}^T. \tag{5.4} $$

The scene graph, shown in Figure 5.5, is depicted at the *rotational velocity* level for the special case of rotational velocity $\boldsymbol{\omega}$ (in pale green) and a general vector $\boldsymbol{c}$ (in pale red). This distinction becomes crucial when deriving the relation between the moving frames $[mf_1]$ and $[mf_2]$ at the rotational velocity level, where a loop is necessary. The special case of rotational velocity allows us to close the loop, highlighted in yellow in Figure 5.5, to determine ${}^{mf_2}_{mf_1}\boldsymbol{\omega}_{mf_1}$.

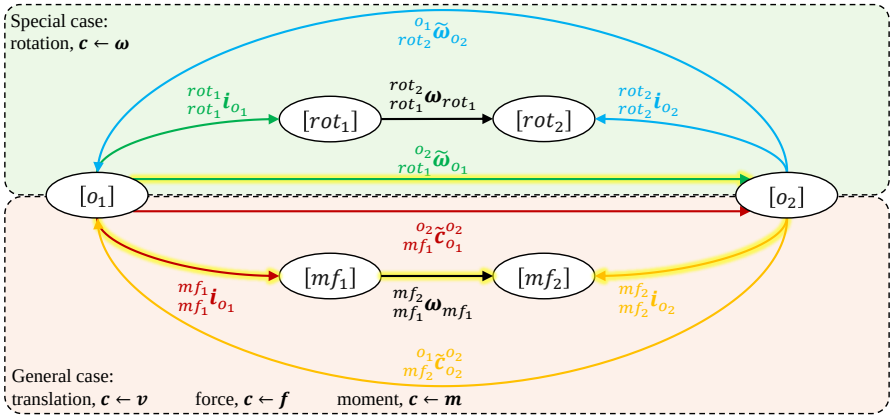


Figure 5.5: Scene graph at *rotational velocity* level for orientation frames, including object frames and FS frames. Vector invariants in two viewpoints explained for the rotational velocity $\boldsymbol{\omega}$ as the special case (in pale green) and a general vector $\boldsymbol{c}$ (in pale red) which could be translational velocity $\boldsymbol{v}$, force $\boldsymbol{f}$, or moment $\boldsymbol{m}$. The highlighted yellow loop is used in (5.8).

The following relation can be written to change the reference frame for object invariant from $[mf_2]$ to $[mf_1]$, using (E.20), (5.1), and (5.3).

$$ {}^{o_2}_{mf_1}\widetilde{\boldsymbol{c}}_{o_1} = -{}^{mf_2}_{mf_1}\boldsymbol{R}_{mf_1}\, {}^{o_1}_{mf_2}\widetilde{\boldsymbol{c}}_{o_2} \tag{5.5} $$

This relation can be further rewritten by applying the definitions of the object invariant from both viewpoints.

$$ \begin{pmatrix} c^{mf_1} \\ 0 \\ 0 \end{pmatrix} = - \begin{pmatrix} -1 & 0 & 0 \\ 0 & -\mathrm{c}\phi_c & \mathrm{s}\phi_c \\ 0 & \mathrm{s}\phi_c & \mathrm{c}\phi_c \end{pmatrix} \begin{pmatrix} c^{mf_2} \\ 0 \\ 0 \end{pmatrix} \tag{5.6} $$

Thus, the following relation holds for the object invariants across two viewpoints. This is expected based on the relation between ${}^{mf_1}\boldsymbol{e}_{t_{o_1}}$ and ${}^{mf_2}\boldsymbol{e}_{t_{o_1}}$, as shown in (E.8), and the definition of the object invariant, as given in (2.16).

$$ c^{mf_1} = c^{mf_2} \tag{5.7} $$

To investigate the relations between the moving frame invariants for a general vector $\boldsymbol{c}$, we begin with the following equation derived from the highlighted yellow loop in the scene graph shown in Figure 5.5. Note that the object invariant for rotation ${}^{o_2}_{rot_1}\widetilde{\boldsymbol{\omega}}_{o_1}$ is included in the loop.

$$ {}^{mf_2}_{mf_1}\boldsymbol{\omega}_{mf_1} = -\,{}^{mf_1}_{mf_1}\boldsymbol{i}_{o_1} + {}^{rot_1}_{mf_1}\boldsymbol{R}_{mf_1}\,{}^{o_2}_{rot_1}\widetilde{\boldsymbol{\omega}}_{o_1} + {}^{mf_2}_{mf_1}\boldsymbol{R}_{mf_1}\,{}^{mf_2}_{mf_2}\boldsymbol{i}_{o_2} \tag{5.8} $$

According to the results in Appendix E, the term ${}^{mf_2}_{mf_1}\boldsymbol{\omega}_{mf_1}$ can be replaced by $(\phi_c' \;\; 0 \;\; 0)^T$. By substituting the expressions from equations (E.20), (5.1), (5.2), and (5.4) for ${}^{mf_2}_{mf_1}\boldsymbol{R}_{mf_1}$, ${}^{o_2}_{rot_1}\widetilde{\boldsymbol{\omega}}_{o_1}$, ${}^{mf_1}_{mf_1}\boldsymbol{i}_{o_1}$, and ${}^{mf_2}_{mf_2}\boldsymbol{i}_{o_2}$, respectively, we obtain the following relation.

$$ \begin{pmatrix} \phi_c' \\ 0 \\ 0 \end{pmatrix} = -\begin{pmatrix} \omega_\tau^{mf_1} \\ 0 \\ \omega_\kappa^{mf_1} \end{pmatrix} + {}^{rot_1}_{mf_1}\boldsymbol{R}_{mf_1}\begin{pmatrix} \omega^{rot_1} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 & 0 \\ 0 & -\mathrm{c}\phi_c & \mathrm{s}\phi_c \\ 0 & \mathrm{s}\phi_c & \mathrm{c}\phi_c \end{pmatrix}\begin{pmatrix} \omega_\tau^{mf_2} \\ 0 \\ \omega_\kappa^{mf_2} \end{pmatrix} \tag{5.9} $$

We can assume the following relation, which transforms the object invariant from $[rot_1]$ to $[mf_1]$:

$$ \begin{pmatrix} \omega_t \\ \omega_n \\ \omega_b \end{pmatrix} = {}^{rot_1}_{mf_1}\boldsymbol{R}_{mf_1}\begin{pmatrix} \omega^{rot_1} \\ 0 \\ 0 \end{pmatrix}, \tag{5.10} $$

where ω^{rot_1} is the object invariant in the FS frame $[mf_1]$ derived from data expressed in the first viewpoint. Substituting (5.10) into (5.9) yields the following relations between the moving frame invariants.

$$ \phi_c' = -\omega_\tau^{mf_1} + \omega_t - \omega_\tau^{mf_2} \tag{5.11} $$

$$ \omega_\kappa^{mf_1} = \omega_b + \omega_\kappa^{mf_2}\mathrm{c}\phi_c \tag{5.12} $$

Note that for rotational FS frames, i.e., $\boldsymbol{c} = \boldsymbol{\omega}$, the relations between vector invariants simplify. In this special case, the binormal axes of these frames coincide, meaning $\phi_c = 0$, as shown in Figure 5.2b. Consequently, the relations between the moving frame invariants reduce to those given in (5.13) and (5.14), since ${}^{rot_1}_{mf_1}\boldsymbol{R}_{mf_1} = \boldsymbol{I}_3$ holds with $mf_1 = rot_1$ in this special case.

$$ \omega_\tau^{rot_1} = \omega^{rot_1} - \omega_\tau^{rot_2} \tag{5.13} $$

$$ \omega_\kappa^{rot_1} = \omega_\kappa^{rot_2} \tag{5.14} $$

The vector invariants exhibit two types of singularities, as discussed in Section 2.3.1. If the first type of singularity occurs in one viewpoint, whether for rotation or a general vector, it will also appear in the other viewpoint. The second

type of singularity is likewise present in both viewpoints for rotation, but it is not necessarily present in both viewpoints for a general vector. This can be understood by comparing (E.15) and (E.17), which describe the unit vectors of the moving frames in two viewpoints, or (F.13) and (F.20), which represent the moving frame invariant ω_κ in two viewpoints.

5.3.2 Screw Invariants

Table 5.4 lists screw invariants in two viewpoints, using the notation introduced in Chapter 2. In contrast to Table 2.3, where the notation is simplified for conciseness, this table retains the full notation for completeness and precision. Similar to the vector invariants, screw invariants also have object invariant $\widetilde{\boldsymbol{\mathcal{I}}}$ and moving frame invariant $\dot{\boldsymbol{\iota}}$ at each viewpoint. The object invariant relates to the object since a and b represent the magnitude of the directional component and moment component of the screw. The moving frame invariant relates to the motion of the moving frame since it models the first-order kinematics of the moving frame.

Table 5.4: Screw invariants with complete notation in two viewpoints.

Screw	Viewpoint	Object Invariant	Moving Frame Invariant
Screw $\boldsymbol{s}$	$\mathcal{O}_1$	${}^{o_2}_{mf_1}\widetilde{\boldsymbol{\mathcal{I}}}^{mf_1}_{o_1} = (a^{mf_1}\ 0\ 0\ b^{mf_1}\ 0\ 0)T$	${}^{mf_1}_{mf_1}\dot{\boldsymbol{\iota}}^{mf_1}_{o_1} = (\omega_\tau^{mf_1}\ 0\ \omega_\kappa^{mf_1}\ v_t^{mf_1}\ 0\ v_b^{mf_1})^T$
	$\mathcal{O}_2$	${}^{o_1}_{mf_2}\widetilde{\boldsymbol{\mathcal{I}}}^{mf_2}_{o_2} = (a^{mf_2}\ 0\ 0\ b^{mf_2}\ 0\ 0)T$	${}^{mf_2}_{mf_2}\dot{\boldsymbol{\iota}}^{mf_2}_{o_2} = (\omega_\tau^{mf_2}\ 0\ \omega_\kappa^{mf_2}\ v_t^{mf_2}\ 0\ v_b^{mf_2})^T$
Twist $\boldsymbol{t}$	$\mathcal{O}_1$	${}^{o_2}_{isam_1}\widetilde{\boldsymbol{t}}^{isam_1}_{o_1} = (\omega^{isam_1}\ 0\ 0\ v^{isam_1}\ 0\ 0)T$	${}^{isam_1}_{isam_1}\dot{\boldsymbol{\iota}}^{isam_1}_{o_1} = (\omega_\tau^{isam_1}\ 0\ \omega_\kappa^{isam_1}\ v_t^{isam_1}\ 0\ v_b^{isam_1})^T$
	$\mathcal{O}_2$	${}^{o_1}_{isam_2}\widetilde{\boldsymbol{t}}^{isam_2}_{o_2} = (\omega^{isam_2}\ 0\ 0\ v^{isam_2}\ 0\ 0)T$	${}^{isam_2}_{isam_2}\dot{\boldsymbol{\iota}}^{isam_2}_{o_2} = (\omega_\tau^{isam_2}\ 0\ \omega_\kappa^{isam_2}\ v_t^{isam_2}\ 0\ v_b^{isam_2})^T$
Wrench $\boldsymbol{w}$	$\mathcal{O}_1$	${}^{o_2}_{isaw_1}\widetilde{\boldsymbol{w}}^{isaw_1}_{o_1} = (f^{isaw_1}\ 0\ 0\ m^{isaw_1}\ 0\ 0)T$	${}^{isaw_1}_{isaw_1}\dot{\boldsymbol{\iota}}^{isaw_1}_{o_1} = (\omega_\tau^{isaw_1}\ 0\ \omega_\kappa^{isaw_1}\ v_t^{isaw_1}\ 0\ v_b^{isaw_1})^T$
	$\mathcal{O}_2$	${}^{o_1}_{isaw_2}\widetilde{\boldsymbol{w}}^{isaw_2}_{o_2} = (f^{isaw_2}\ 0\ 0\ m^{isaw_2}\ 0\ 0)T$	${}^{isaw_2}_{isaw_2}\dot{\boldsymbol{\iota}}^{isaw_2}_{o_2} = (\omega_\tau^{isaw_2}\ 0\ \omega_\kappa^{isaw_2}\ v_t^{isaw_2}\ 0\ v_b^{isaw_2})^T$

We assume that screw invariants in two viewpoints are available for a general screw $\boldsymbol{\mathcal{I}}$ as in the first line of Table 5.4. The object and moving frame invariants in the first viewpoint are given as follows:

$$ {}^{o_2}_{mf_1}\widetilde{\boldsymbol{\mathcal{I}}}^{mf_1}_{o_1} = \begin{pmatrix} a^{mf_1} & 0 & 0 & b^{mf_1} & 0 & 0 \end{pmatrix}^T \tag{5.15} $$

$$ {}^{mf_1}_{mf_1}\dot{\boldsymbol{\iota}}^{mf_1}_{o_1} = \begin{pmatrix} \omega_\tau^{mf_1} & 0 & \omega_\kappa^{mf_1} & v_t^{mf_1} & 0 & v_b^{mf_1} \end{pmatrix}^T, \tag{5.16} $$

while the object and moving frame invariants in the second viewpoint are as follows:

$$ {}^{o_1}_{mf_2}\widetilde{\boldsymbol{\mathcal{I}}}^{mf_2}_{o_2} = \begin{pmatrix} a^{mf_2} & 0 & 0 & b^{mf_2} & 0 & 0 \end{pmatrix}^T \tag{5.17} $$

$$ {}^{mf_2}_{mf_2}\dot{\boldsymbol{\iota}}^{mf_2}_{o_2} = \begin{pmatrix} \omega_\tau^{mf_2} & 0 & \omega_\kappa^{mf_2} & v_t^{mf_2} & 0 & v_b^{mf_2} \end{pmatrix}^T. \tag{5.18} $$

The scene graph is depicted in Figure 5.6 at *twist* level for the special case of twist (in pale green) and a general screw (in pale red). However, in this subsection, the general case is limited to a wrench. The special case of twist allows us to close the loop, highlighted in yellow in Figure 5.6, to determine ${}^{mf_2}_{mf_1}\boldsymbol{t}^{mf_1}_{mf_1}$.

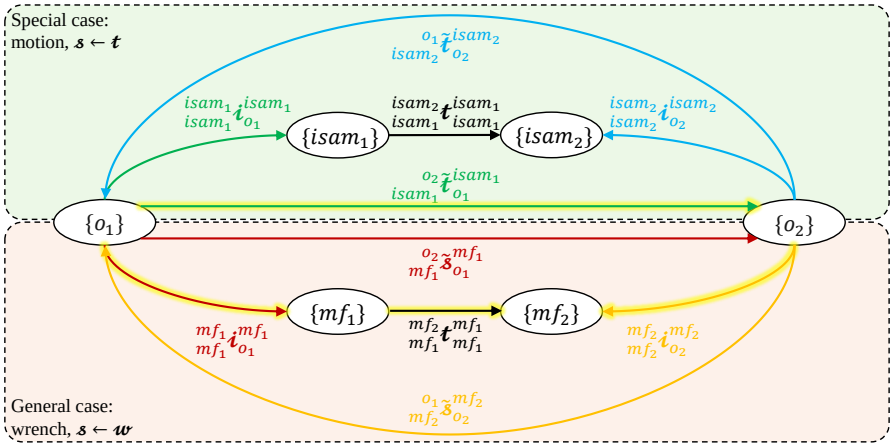


Figure 5.6: Scene graph at *twist* level for frames, including object frames and ISA frames. Screw invariants in two viewpoints explained for twist $\boldsymbol{t}$ as the special case (in pale green) and a general screw $\boldsymbol{\mathfrak{z}}$ which could be wrench $\boldsymbol{w}$ (in pale red). The highlighted yellow loop is used in (5.23) to study the relation between the moving frames $\{mf_1\}$ and $\{mf_2\}$.

The following relation can be written which is the transformation of object invariant from $\{mf_2\}$ to $\{mf_1\}$ by a screw transformation matrix, using (F.24), (5.15), and (5.17).

$$ {}^{o_2}_{mf_1}\widetilde{\boldsymbol{\mathfrak{z}}}^{mf_1}_{o_1} = -\,{}^{mf_2}\boldsymbol{S}_{mf_1}\,{}^{o_1}_{mf_2}\widetilde{\boldsymbol{\mathfrak{z}}}^{mf_2}_{o_2} \tag{5.19} $$

This relation can be further rewritten by applying the definitions of the object invariant from both viewpoints.

$$ \begin{pmatrix} a^{mf_1} \\ 0 \\ 0 \\ b^{mf_1} \\ 0 \\ 0 \end{pmatrix} = - \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\mathrm{c}\phi & \mathrm{s}\phi & 0 & 0 & 0 \\ 0 & \mathrm{s}\phi & \mathrm{c}\phi & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & -p_t\mathrm{s}\phi & -p_t\mathrm{c}\phi & 0 & -\mathrm{c}\phi & \mathrm{s}\phi \\ 0 & -p_t\mathrm{c}\phi & p_t\mathrm{s}\phi & 0 & \mathrm{s}\phi & \mathrm{c}\phi \end{pmatrix} \begin{pmatrix} a^{mf_2} \\ 0 \\ 0 \\ b^{mf_2} \\ 0 \\ 0 \end{pmatrix} \tag{5.20} $$

This results in the following relations, which are anticipated based on the relation between ${}^{mf_1}\boldsymbol{e}_{t\,o_1}$ and ${}^{mf_2}\boldsymbol{e}_{t\,o_1}$, as shown in (E.8), and the definition of

the object invariant, as given in (2.37).

$$a^{mf_1} = a^{mf_2} \tag{5.21}$$

$$b^{mf_1} = b^{mf_2} \tag{5.22}$$

To explore the relations between the moving frames $\{mf_1\}$ and $\{mf_2\}$, we can start with the following equation based on the highlighted yellow loop in the scene graph shown in Figure 5.6.

$${}^{mf_2}_{mf_1}\boldsymbol{t}^{mf_1}_{mf_1} = -\,{}^{mf_1}_{mf_1}\boldsymbol{i}^{mf_1}_{o_1} + {}^{isam_1}\boldsymbol{S}_{mf_1}\,{}^{o_2}_{isam_1}\widetilde{\boldsymbol{t}}^{isam_1}_{o_1} + {}^{mf_2}\boldsymbol{S}_{mf_1}\,{}^{mf_2}_{mf_2}\boldsymbol{i}^{mf_2}_{o_2} \tag{5.23}$$

By substituting the expressions from equations (F.24), (5.15), (5.16), and (5.18) for ${}^{mf_2}\boldsymbol{S}_{mf_1}$, ${}^{o_2}_{isam_1}\widetilde{\boldsymbol{t}}^{isam_1}_{o_1}$, ${}^{mf_1}_{mf_1}\boldsymbol{i}^{mf_1}_{o_1}$, and ${}^{mf_2}_{mf_2}\boldsymbol{i}^{mf_2}_{o_2}$, respectively, we can find the following relation.

$$\begin{aligned}{}^{mf_2}_{mf_1}\boldsymbol{t}^{mf_1}_{mf_1} = &-\begin{pmatrix}\omega_\tau^{mf_1}\\0\\\omega_\kappa^{mf_1}\\v_t^{mf_1}\\0\\v_b^{mf_1}\end{pmatrix} + {}^{isam_1}\boldsymbol{S}_{mf_1}\begin{pmatrix}\omega^{isam_1}\\0\\0\\v^{isam_1}\\0\\0\end{pmatrix} + \\ &\begin{pmatrix}-1&0&0&0&0&0\\0&-\mathrm{c}\phi&\mathrm{s}\phi&0&0&0\\0&\mathrm{s}\phi&\mathrm{c}\phi&0&0&0\\0&0&0&-1&0&0\\0&-p_t\mathrm{s}\phi&-p_t\mathrm{c}\phi&0&-\mathrm{c}\phi&\mathrm{s}\phi\\0&-p_t\mathrm{c}\phi&p_t\mathrm{s}\phi&0&\mathrm{s}\phi&\mathrm{c}\phi\end{pmatrix}\begin{pmatrix}\omega_\tau^{mf_2}\\0\\\omega_\kappa^{mf_2}\\v_t^{mf_2}\\0\\v_b^{mf_2}\end{pmatrix}\end{aligned} \tag{5.24}$$

According to the results in Appendices E and F, the term ${}^{mf_2}_{mf_1}\boldsymbol{t}^{mf_1}_{mf_1}$ can be replaced as follows:

$${}^{mf_2}_{mf_1}\boldsymbol{t}^{mf_1}_{mf_1} = \begin{pmatrix}\phi_j' & 0 & 0 & p_t' & 0 & 0\end{pmatrix}^T, \tag{5.25}$$

and we can assume the following relation, which transforms the object invariant from $\{isam_1\}$ to $\{mf_1\}$.

$$\begin{pmatrix}\omega_t & \omega_n & \omega_b & v_t & v_n & v_b\end{pmatrix}^T = {}^{isam_1}\boldsymbol{S}_{mf_1}\begin{pmatrix}\omega^{isam_1} & 0 & 0 & v^{isam_1} & 0 & 0\end{pmatrix}^T \tag{5.26}$$

Then, by substituting (5.25) and (5.26) into (5.24), we can derive the relation between the moving frames $\{mf_1\}$ and $\{mf_2\}$.

$$\begin{pmatrix} \phi_\jmath' \\ 0 \\ 0 \\ p_t' \\ 0 \\ 0 \end{pmatrix} = - \begin{pmatrix} \omega_\tau^{mf_1} \\ 0 \\ \omega_\kappa^{mf_1} \\ v_t^{mf_1} \\ 0 \\ v_b^{mf_1} \end{pmatrix} + \begin{pmatrix} \omega_t \\ \omega_n \\ \omega_b \\ v_t \\ v_n \\ v_b \end{pmatrix} + \begin{pmatrix} -\omega_\tau^{mf_2} \\ \omega_\kappa^{mf_2}\ \mathrm{s}\phi_\jmath \\ \omega_\kappa^{mf_2}\ \mathrm{c}\phi_\jmath \\ -v_t^{mf_2} \\ -p_t\ \mathrm{c}\phi_\jmath\ \omega_\kappa^{mf_2} + v_b^{mf_2}\ \mathrm{s}\phi_\jmath \\ p_t\ \mathrm{s}\phi_\jmath\ \omega_\kappa^{mf_2} + v_b^{mf_2}\ \mathrm{c}\phi_\jmath \end{pmatrix} \tag{5.27}$$

By inspecting (5.27), we obtain the following relations between the moving frame invariants derived from two viewpoints.

$$\phi_\jmath' + \omega_\tau^{mf_1} + \omega_\tau^{mf_2} = \omega_t \tag{5.28}$$

$$\omega_n + \omega_\kappa^{mf_2}\ \mathrm{s}\phi_\jmath = 0 \tag{5.29}$$

$$\omega_\kappa^{mf_1} = \omega_b + \omega_\kappa^{mf_2}\ \mathrm{c}\phi_\jmath \tag{5.30}$$

$$p_t' + v_t^{mf_1} + v_t^{mf_2} = v_t \tag{5.31}$$

$$v_n - p_t\ \mathrm{c}\phi_\jmath\ \omega_\kappa^{mf_2} + v_b^{mf_2}\ \mathrm{s}\phi_\jmath = 0 \tag{5.32}$$

$$v_b^{mf_1} = v_b + p_t\ \mathrm{s}\phi_\jmath\ \omega_\kappa^{mf_2} + v_b^{mf_2}\ \mathrm{c}\phi_\jmath \tag{5.33}$$

Note that in the special case of ISA frames of motion, the relations between moving frame invariants simplify. This is because the binormal axes of these frames coincide, i.e., $\phi_\jmath = 0$, and their origins also coincide, i.e., $p_t = 0$, as shown in Figure 5.4b. Under these conditions, the relations between the moving frame invariants reduce to those given in (5.34)-(5.37) since ${}^{isam_1}\boldsymbol{S}_{mf_1} = \boldsymbol{I}_6$ with $mf_1 = isam_1$. Moreover, all the components of (5.29) and (5.32) vanish. Therefore, the following simplified relations hold.

$$\omega_\tau^{isam_1} + \omega_\tau^{isam_2} = \omega^{isam_1} \tag{5.34}$$

$$\omega_\kappa^{isam_1} = \omega_\kappa^{isam_2} \tag{5.35}$$

$$v_t^{isam_1} + v_t^{isam_2} = v_t^{isam_1} \tag{5.36}$$

$$v_b^{isam_1} = v_b^{isam_2} \tag{5.37}$$

Similar to the vector invariants, the screw invariants also exhibit two types of singularities, as discussed in Section 2.4.1. If the first type of singularity occurs in one viewpoint, whether for motion or a general screw, it will also appear in the other viewpoint. The second type of singularity is likewise present in both viewpoints for motion, but it is not necessarily present in both viewpoints for a general screw. This can be understood by comparing (E.15) and (E.17), which describe the unit vectors of the moving frames in two viewpoints, or (F.13) and (F.20), which represent the moving frame invariant ω_κ in two viewpoints.

5.4 Conclusion

This chapter provides a comprehensive investigation of the relation between invariants and their associated moving frames derived from data expressed in two viewpoints. The relations for both vector and screw invariants are established.

For FS frames, it is shown that the relative orientation between the moving frames of rotational motion exhibits a 180° flip while for other cases such as translational motion, force, and moment, the relative orientation involves a rotation around the tangent axis.

For ISA frames, it is discussed that the relative orientation of moving frames follows the same formulation as that developed for the FS frames. The relative position of ISA frames of motion coincide while the one of wrench does not coincide and they are displaced along the tangent axis.

For screw invariants of motion and vector invariants of rotation, the relation between moving frame invariants derived from two viewpoints has a simple form. Thus, we can calculate invariants in two viewpoints more efficiently. For instance, if the invariants are computed in one viewpoint (e.g., via an OCP), they can be readily transformed to the other viewpoint without recalculating.

In contrast, for screw invariants of wrench and vector invariants of translation/force/moment, the relation between moving frame invariants derived from two viewpoints has a complex form. Then, these invariants are needed to be calculated separately in two viewpoints.

Singularities exhibit distinct behaviors across the viewpoints. If the first type of singularity occurs in one viewpoint for any types of screw invariants or vector invariants, it will also appear in the other viewpoint. Similarly, the second type of singularity in screw invariants of motion or vector invariants of rotation is always present in both viewpoints.

However, for screw invariants of wrench and vector invariants of translation/-force/moment, the occurrence of the second type of singularity may vary between viewpoints. For example, in the 3-D Contour Following experiment discussed in Section 2.5, the ISA of wrench exhibits the second type of singularity in the tool (second) viewpoint but not in the world (first) viewpoint, as illustrated in Figures 2.10b and 2.10c.

Although the relations between the invariants derived from two viewpoints are presented, this chapter is limited to a mathematical representation and an exploration of the effects of viewpoint selection. It does not include a practical application to demonstrate the utility of these findings. A potential direction

for future work could involve applying these relations to bi-manual tasks, where a task might be modeled from two viewpoints. However, the practical benefits of such an approach remain uncertain and would require thorough investigation to assess its potential added value.

Chapter 6

Conclusion

This chapter provides a comprehensive summary of the main contributions of the thesis, emphasizing the key findings. Additionally, it addresses the limitations of the current research, acknowledging areas where further refinement or exploration is needed. Finally, the chapter outlines potential directions for future research, suggesting how the work can be extended or applied to new challenges.

6.1 Contributions

In this thesis, we propose methods aimed at enabling robots to learn and perform contact-rich tasks effectively. These tasks, which always involve physical interaction with environment, such as insertion, assembly, or manipulation, are common in many industrial applications. Developing robust solutions for such tasks is crucial, especially in light of the challenges outlined in Section 1, including the growing shortage of skilled labor and the increasing demand for flexible automation.

The proposed methods contribute toward addressing these challenges by allowing robots to adapt to variations within a class of tasks or products without the need for exhaustive reprogramming or manual intervention. For example, a robot trained to open one type of door should be able to generalize to similar doors with different handle positions or hinge distances. By focusing on generalizable and adaptable learning strategies, the proposed methods support deployment of robotic solutions where task conditions may vary but still belong to the same

general category.

1. A Novel Method for Modeling Contact Tasks Using Invariant Descriptors

Chapter 2 introduces invariant descriptors for wrench trajectories, analogous to motion invariant descriptors for rigid bodies and point trajectories. Thus, in general, two types of invariant descriptors can be calculated: 3-D vector invariant descriptors (translational velocities, rotational velocities, forces, and moments) or 6-D screw invariant descriptors (twists and wrenches). A novel OCP is proposed within this chapter, designed to be intuitive to tune, and its effectiveness is validated in the context of LfD through two tasks: 3-D Contour Following and Peg on Hole Alignment. Modeling each task requires only a single demonstration trial, similar to [107].

For the 3-D Contour Following task, an extensive set of experiments is conducted to evaluate the invariance of these descriptors under changes in body and world reference frames. The changes applied to the body reference frame are illustrated in Figure 2.5, while the changes applied to the world reference frame are detailed in Table 2.4. The OCP is tuned only once as described in Section 2.5.4, and then is applied consistently across all demonstration trials. Additionally, the results are compared to reference values for the limit case to assess the model's repeatability when differences exist in the coordinates of the recorded data.

While the properties of the invariants are thoroughly analyzed in the context of the 3-D Contour Following task, a Peg on Hole Alignment experiment is also presented. This experiment confirms the simplicity of the tuning procedure and highlights the practicality of using these descriptors for task modeling.

2. A Novel Method for Modeling and Generalizing Contact-Rich Tasks Using Screw Theory and Constraint-Based Control Strategy

The selection of the task frame has traditionally been left to the knowledge and intuition of the expert responsible for robot programming. However, in Chapter 3, a novel insight is introduced to uncover the implicit assumptions underlying task frame selection. These assumptions are formalized as hypotheses and evaluated within the context of LfD where quite a few demonstration trials are required to derive the task frame from the recorded data.

To test these hypotheses, seven diverse tasks were selected, including a range of characteristics: some tasks involved dominant translational motions, while others featured dominant rotational motions; similarly, some tasks emphasized dominant forces, while others required dominant moments.

The experimental results validate the hypotheses:

- The origin of the task frame is selected to minimize translational velocity or moment, or to ensure minimal variation in these quantities.
- The orientation of the task frame is chosen to maximize decoupling between the directions of force/moment and translational/rotational velocity.

This insight provides a systematic foundation for task frame selection, reducing reliance on expert intuition for task frame design in contact-rich robotic applications.

The aforementioned insight is achieved while formulating a method for optimal task frame derivation. This method, depicted in Figure 3.6, is automatic, data-driven, and requires no thresholds or additional parameters throughout the procedure.

To demonstrate the effectiveness of the findings, the derived task frames for all tasks are compared in detail with those designed by the expert. The results, shown in Table 3.4, confirm that the derived task frame is close to the one designed by the expert.

Additionally, the derived task frame is used to perform the demonstrated tasks with a UR10e robot. The results validate the effectiveness of the method. Since a control strategy is also designed and employed for task performance, it is worth briefly mentioning its features. The applied control strategy is a constraint-based controller, which includes the following characteristics:

- Desired values of pose and wrench are followed through constraints.
- These constraints are inherently conflicting. The conflict is resolved according to the assigned weights.
- Twist is incorporated as feedforward in the constraints.
- There is no need to adjust the constraints for each task, as all constraints related to position, rotation, translational velocity, rotational velocity, force, and moment are always active.
- Pose constraints between the tool and the environment as well as the desired pose values are expressed in the task frame using similarity transformation, highlighting the practical advantages of using similarity transformation.

3. A Novel Method for Estimating FT Sensor Offsets and Gravity Parameters Using only Wrench Measurements

To accurately compensate for the tool's weight, the FT sensor offsets and gravity parameters, i.e. mass and center of mass (COM), are required. While various methods exist to estimate these unknowns, they often rely on using both orientation and wrench information. In contrast, the method proposed in Chapter 4 eliminates the need for orientation measurements, making it particularly well-suited for applications where orientation is unavailable or unreliable.

The proposed method is compared to a baseline that utilizes both wrench and accurate orientation measurements, whereas our method relies only on wrench data. The results show that while both methods achieve good estimations, the proposed method consistently outperforms the baseline across all real-world experiments within the excitation range.

4. A thorough Discussion on the Invariant Descriptors Derived from Different Viewpoints

Typically, motion of an object is expressed relative to a fixed frame, commonly referred to as world frame, using a frame attached to the object, referred to as body frame. In this context, invariants are calculated based on the coordinates of the body frame expressed with respect to the world frame. However, in force analysis, it can sometimes be more appropriate to analyze invariants either in the body frame or in the world frame, depending on the specific application.

This dual perspective forms the core contribution of Chapter 5, where the relations between invariants in the two frames are derived. It is shown that these invariants are not identical across the two viewpoints. While the practical utility of these relations is not explicitly validated within this chapter, they provide valuable insights for the experimental design discussed in Section 2.5.2 and for the task frame selection procedure outlined in Section 3.4.

By analyzing invariants from both viewpoints, this work lays a foundation for understanding how viewpoint selection influences force and motion invariants by offering a thorough analysis and investigation.

6.2 Limitations

1. Segmentation of Contact-Rich Task

Segmentation involves identifying meaningful segments within a task where there are no abrupt changes in the quantities of interest. Each segment has a specific goal, and together they contribute to accomplishing the overall objective of the task. However, identifying such segments is not straightforward. In some studies, such as [80], segmentation is performed manually. To overcome the limitations of manual segmentation, various automated techniques have been proposed that detect abrupt changes in sensory data. For example, [100, 57, 64, 63] leverage sudden variations in motion and/or wrench signals, often in combination with other modalities, to segment demonstrations into meaningful components. Additionally, a screw-based trajectory-shape descriptor was introduced in a parallel work in our research group [101], enabling self-supervised segmentation by capturing geometric characteristics of motion. This descriptor contains information of the magnitude and local change in direction of velocity that facilitates data segmentation.

While these advanced segmentation techniques are promising, the work presented in Chapters 2, 3, and 4 assumes that the segment of interest is already available, using one of the methods mentioned above. In practice, we identified these segments using simple threshold-based methods for rotational/translational velocity and contact force/moment. While these approaches sufficed for our analysis, a more robust segmentation method could improve the precision of identifying contact tasks, thereby resulting in more accurate modeling. Moreover, such an approach could integrate additional segments, such as approach or retraction phases, to create a comprehensive set of skills. Clearly, considering these segments would require incorporating vision into the robot setup to localize the object of interest. This would address the limitations of the current version, which focuses exclusively on the contact segment.

A potential direction for future work could involve developing a segmentation method based on the uncertainty associated with origin candidates presented in Chapter 3. Since the process of task frame's origin determination begins with eight candidates, as illustrated in Figure 3.6, a method could be designed to identify distinct segments within a moving window based on combining and/or comparing the uncertainties of the candidates from two viewpoints. Within these segments, the task frame may adopt different viewpoints or vary in its origin and orientation.

2. Compliance in Contact-Rich Task Modeling

Chapter 3 presents a framework to model a variety of contact-rich tasks according to motion and wrench data recorded during human demonstrations. While we have made efforts to extract as much information as possible from the demonstrated data, there is still potential for further investigation. For instance, we extracted desired values for pose, twist, and wrench within the derived task frame. However, the relation between motion and wrench could be better explained through (varying) compliance. This aspect represents a gap in the current model. Although a model is developed based on motion and wrench data, the relation between the motion and wrench is currently simplified by assuming a diagonal compliance matrix, as described in Section 3.5.5 and with the values reported in Table 3.3, which may not fully capture the complexity of the interactions.

3. Task Frame Definition and Generalization Across New Objects

Chapter 3 assumes that the task frame is expressed relative to arbitrarily selected object frames (i.e., tool or world frames) defined within their corresponding viewpoints. The approach presented in Section 3.6.2 enables the task frame to adapt to variations within the same object category, such as accommodating a longer bottle opener or a smaller peg. This adaptability is achieved because, in both cases, the tool frames were chosen strategically, with their origins attached to a specific feature of the tool.

To generalize this process for systematically identifying the same feature on new objects, an automatic method for feature identification is required. Once this feature is accurately identified, the object frame can be correctly assigned to either the tool or the environment for a new tool or setting. Consequently, the task frame's location and orientation can be adjusted automatically. For further details and examples, refer to Appendix J.

4. Task Frame Derivation for Combined Rotational and Translational Tasks

In Chapter 3, a novel approach is introduced to model contact-rich tasks characterized by dominant rotational or translational motions. Leveraging the duality between motion and wrench, tasks can also exhibit dominant forces or moments.

However, certain tasks, such as opening an airplane tray table, inherently allow combining both rotation and translation. For instance, a user might

simultaneously rotate and translate the tray, introducing significant uncertainty into the task modeling. To resolve this, it is advisable to decompose such tasks into sequential segments, first addressing rotation and then translation, or vice versa, thereby reducing complexity and improving model accuracy.

5. Human Demonstration Quality

Another limitation of the approaches presented in this thesis is their reliance on high-quality demonstrations to build reliable models. A high-quality demonstration refers to a demonstration performed by an expert operator that captures the essential features of a task and ensures the achievement of the task's goal. However, as in many studies, including this work, it is commonly assumed that the operator is capable of providing such high-quality demonstrations [89]. Specifically:

- In Chapter 2, the model relies on a single demonstration trial. Therefore, the demonstrator must perform the task carefully, ensuring that all critical factors necessary for task completion are included. For example, when inserting a peg into a hole, rotating the peg around its symmetry axis may occur during the demonstration. However, such rotations are not essential to the task's primary goal, which is achieving insertion.
- In Chapter 3, the focus is on estimating the task frame based on the certainty derived from the provided data. This method assumes that demonstration trials are repeatable and effectively excite the essential motions and wrenches required for the task. For example, when following a 3-D contour with a specific contact force, demonstrators should avoid applying unnecessary moments. Introducing such non-essential elements could result in an incorrect task frame, potentially rendering it unable to perform the intended task.
- In Chapter 4, it is critical that the demonstration data include sufficient orientation excitation, at least comparable to the range of orientation expected during the actual task. This ensures better estimation for sensor offsets and gravity parameters.

From a broader perspective, these considerations highlight the importance of evaluating the quality of the demonstrated data. Establishing a method to identify and exclude poor-quality trials or unreliable parts of the data would significantly enhance the robustness of the approach.

6. Systematic Evaluation of Task Performance

Chapter 3 presents a framework to perform contact-rich tasks while Chapter 4 presents a method to estimate sensor offsets and gravity parameters. Although the results, as summarized in Tables 3.5 and 4.3, are satisfactory, a systematic evaluation of the task performance or the accuracy of the unknowns estimation is absent. This implies:

- For task performance, we lack insight into the quality of the performed task. However, as shown in Section 3.6.2, results can be improved by analyzing the data recorded during the robot's performance. It is shown that the iterative refinement of the task frame derivation could address calibration errors between the demonstration setup and the robot's operational environment, including differences in compliance.
- For sensor offsets and gravity parameters estimation, we also lack insight on the accuracy of the results. But incorporating uncertainty of the estimated COM can guide and evaluate the orientation excitation.

Therefore, the current version of this work appears to lack a crucial evaluation step. Adding such a step could address this limitation and enhance the overall approach.

6.3 Future Work

1. Relation between ASIP, AVOF, and ISA Frame

The ASIP and AVOF tools introduced in Chapter 3 build extensively on the concept of the average screw axis (ASA) introduced in [10]. While the ASA frame and the ISA frame are intuitively related, their relationship has not been thoroughly investigated in the literature. As shown in Figure 2.14a, the rotation of the peg in the Peg on Hole Alignment task is modeled with a stationary ISA frame. This frame satisfies the proposed constraints related to the tuning parameters (tolerances) incorporated in the OCP, which intuitively corresponds to the ASA frame where the task is modeled with an average frame.

Preliminary observations suggest that the relation between the ISA and ASA can be structured based on the tolerances of the OCP:

- Low tolerance: The ISA exhibits a lot of variations to satisfy strict tolerance constraints.

- Middle-range tolerance: The ISA and ASA exhibit similar behavior, as seen in Figures 2.14a for the peg's motion and Figures 2.14c for the peg-hole contact wrench.
- High tolerance: The ISA closely approximates the reference frame initialized within the OCP, as the high tolerance reduces the need for strict adherence to the constraints.

This relation can also be extended to the connection between min-var ISA and min-var ASA, provided that the entire segment is available in advance. This extension involves subtracting the average screw from all samples to achieve the min-var ISA.

2. Task Frame Derivation in a Moving Window

Another promising direction for future research on task frame derivation is the development of an instantaneous task frame. This approach can be viewed as an instantaneous extension of the task frame derivation method, utilizing data streamed within a moving window. Unlike the method presented in Chapter 3, an instantaneous task frame would be capable of capturing and adapting to local features of a task. For instance, an instantaneous task frame could switch its viewpoint of origin and/or orientation according to specific criteria such as uncertainty. This relates to the first limitation discussed and can be utilized for segmentation. Additionally, an instantaneous task frame with a moving origin and/or orientation can facilitate, making it suitable for automatic tracking applications such as following an unknown contour.

2. Enhancing Task Performance and Failure Diagnosis

As shown in Section 3.6.2, iterative processes have the potential to enhance task performance quality, as evidenced by improvements in RMSE. Building upon this observation, a promising direction for future work would involve the development of a systematic iterative algorithm. Such an algorithm could be further extended to include methods for detecting and diagnosing task failures, which could then be incorporated into an iterative refinement process. This advancement would provide a robust framework to bridge the gap between ideal demonstrations and the inherent variability encountered in real-world applications, thereby improving adaptability and reliability in practical scenarios.

3. Estimating Tool Center Point Frame

In this thesis, the concept of ASIP is introduced in Chapter 3 and is used to estimate the origin of the task frame. It is also applied to wrench measurements in Chapter 4 to estimate the COM. While the results were promising, this concept could similarly be extended to the motion domain with specific applications. For example, the min-var ASIP could be employed to identify the point with minimal variation in translational velocity, which could prove advantageous for tasks such as 3-D Contour Following.

4. Leveraging Invariant Relations in Practice

Another direction for future research involves practical application of the relations between invariants derived from two viewpoints, where the model can leverage the invariance observed from two different viewpoints. Bi-manual tasks, for instance, present a compelling scenario for further investigation.

Additionally, when a singularity exists in one viewpoint but not in the other, the invariants can be calculated in the singular viewpoint with assistance from the non-singular viewpoint. Thus, this approach might have the potential to improve the robustness and applicability of the model in real-world scenarios.

Appendix A

Visualization of Old and New Notations of FS Frames

The relation between the old and new FS frames is visualized in Figure A.1. The old one is used in the earlier publications while the new one is used in the recent publications.

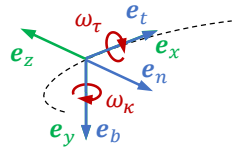


Figure A.1: Visualization of the XYZ frame versus the TNB frame. The XYZ frame shows the old convention used in our group for FS frame in earlier publications while the TNB frame shows the new convention used in our group for FS frame applied in recent publications. This newer convention is also more widely used publicly.

Appendix B

Relation between the Kinematics of the FS Frame and the FS Equation

In differential geometry, the FS equation is usually written as follows:

$$\begin{bmatrix} \boldsymbol{T}' \\ \boldsymbol{N}' \\ \boldsymbol{B}' \end{bmatrix} = \begin{bmatrix} 0 & \omega_\kappa & 0 \\ -\omega_\kappa & 0 & \omega_\tau \\ 0 & -\omega_\tau & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{T} \\ \boldsymbol{N} \\ \boldsymbol{B} \end{bmatrix}, \tag{B.1}$$

with ω_κ the curvature, ω_τ the torsion, and $\boldsymbol{T}$, $\boldsymbol{N}$, $\boldsymbol{B}$ the unit vectors of the FS frame. This equation can be shown to be equal to the kinematics of the FS frame as defined in (2.18):

$$\widetilde{\boldsymbol{R}}' = \widetilde{\boldsymbol{R}} \begin{bmatrix} 0 & -\omega_\kappa & 0 \\ \omega_\kappa & 0 & -\omega_\tau \\ 0 & \omega_\tau & 0 \end{bmatrix}. \tag{r.2.18}$$

In the previous expression, first expand $\widetilde{\boldsymbol{R}}$ into its individual columns using the unit vectors defined in (2.15):

$$\begin{bmatrix} \boldsymbol{e}_t' & \boldsymbol{e}_n' & \boldsymbol{e}_b' \end{bmatrix} = \begin{bmatrix} \boldsymbol{e}_t & \boldsymbol{e}_n & \boldsymbol{e}_b \end{bmatrix} \begin{bmatrix} 0 & -\omega_\kappa & 0 \\ \omega_\kappa & 0 & -\omega_\tau \\ 0 & \omega_\tau & 0 \end{bmatrix}. \tag{B.2}$$

Applying the transpose to both sides of (B.2) results in:

$$\begin{bmatrix} \boldsymbol{e}_t'^T \\ \boldsymbol{e}_n'^T \\ \boldsymbol{e}_b'^T \end{bmatrix} = \begin{bmatrix} 0 & -\omega_\kappa & 0 \\ \omega_\kappa & 0 & -\omega_\tau \\ 0 & \omega_\tau & 0 \end{bmatrix}^T \begin{bmatrix} \boldsymbol{e}_t^T \\ \boldsymbol{e}_n^T \\ \boldsymbol{e}_b^T \end{bmatrix}. \tag{B.3}$$

Since the transpose of a skew-symmetric matrix is equal to the negative of the matrix, (B.3) corresponds to the FS equation in (B.1).

Appendix C

Combining Orientation Estimates

This appendix explains the semantics of the covariance matrix on the orientation as used in this text and discusses Algorithm 4 that combines two orientations estimates $\boldsymbol{R}_1$ with covariance matrix $\boldsymbol{C}_1$ and $\boldsymbol{R}_2$ with covariance $\boldsymbol{C}_2$ into a new estimate $\boldsymbol{R}$ with covariance $\boldsymbol{C}$.

A probabilistic orientation estimate $\hat{\boldsymbol{R}}_i$ can be described by an orientation $\boldsymbol{R}_i$ and a covariance $\boldsymbol{C}_i$ on a probabilistic vector $\hat{\boldsymbol{\delta}}$ using $\hat{\boldsymbol{R}}_i = \exp([\hat{\boldsymbol{\delta}}]_\times)\boldsymbol{R}_i$ with $\hat{\boldsymbol{\delta}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}_i)$. Therefore, $\hat{\boldsymbol{R}}_i$ represents the following distribution:

$$\hat{\boldsymbol{R}}_i \sim P(\boldsymbol{R}|\boldsymbol{R}_i, \boldsymbol{C}_i) = \lambda_i \exp\left(-\tfrac{1}{2}\boldsymbol{\delta}_i^T \boldsymbol{C}_i^{-1} \boldsymbol{\delta}_i\right), \tag{C.1}$$

with

$$\boldsymbol{\delta}_i = \log^\vee\left(\boldsymbol{R}\boldsymbol{R}_i^{-1}\right) \text{ and } \lambda_i = (2\pi)^{-3/2} \det(\boldsymbol{C}_i)^{-1/2}. \tag{C.2}$$

The maximum likelihood estimate $\boldsymbol{R}^*$ given two estimates $\hat{\boldsymbol{R}}_1$ and $\hat{\boldsymbol{R}}_2$ described by $\boldsymbol{R}_1$, $\boldsymbol{C}_1$, $\boldsymbol{R}_2$ and $\boldsymbol{C}_2$ is:

$$\boldsymbol{R}^* = \max_{\boldsymbol{R}} \quad \lambda_1\lambda_2 \exp\left(-\tfrac{1}{2}\boldsymbol{\delta}_1^T \boldsymbol{C}_1^{-1} \boldsymbol{\delta}_1\right) \exp\left(-\tfrac{1}{2}\boldsymbol{\delta}_2^T \boldsymbol{C}_2^{-1} \boldsymbol{\delta}_2\right)$$

Noting that the logarithm is a monotonic function, the above optimization is equivalent to the following minimization:

$$\boldsymbol{R}^* = \min_{\mathbf{R}} \quad \tfrac{1}{2}\boldsymbol{\delta}_1^T \boldsymbol{C}_1^{-1} \boldsymbol{\delta}_1 + \tfrac{1}{2}\boldsymbol{\delta}_2^T \boldsymbol{C}_2^{-1} \boldsymbol{\delta}_2, \tag{C.3}$$

This is a difficult minimization problem due to the non-linear relationships between $\boldsymbol{R}$ and $\boldsymbol{\delta}_i$ in (C.2):

$$\boldsymbol{R}^* = \min_{\boldsymbol{R}} \quad \tfrac{1}{2} \log^\vee \left(\boldsymbol{R}\boldsymbol{R}_1^{-1}\right)^T \boldsymbol{C}_1^{-1} \log^\vee \left(\boldsymbol{R}\boldsymbol{R}_1^{-1}\right) + \\ \tfrac{1}{2} \log^\vee \left(\boldsymbol{R}\boldsymbol{R}_2^{-1}\right)^T \boldsymbol{C}_2^{-1} \log^\vee \left(\boldsymbol{R}\boldsymbol{R}_2^{-1}\right) \tag{C.4}$$

Therefore, a numerical procedure is designed according to the same principles as [54], where we iteratively express all variables with respect to an approximation $\boldsymbol{R}^{(k)}$ of the maximum likelihood $\boldsymbol{R}^*$; then approximate the resulting optimization problem as a quadratic optimization and solve this optimization; after which we update our approximation of $\boldsymbol{R}^*$ to $\boldsymbol{R}^{(k+1)}$ and repeat the procedure.

In iteration step k, expressing the variables in (C.3) with respect to $\boldsymbol{R}^{(k)}$ gives:

$$\boldsymbol{R} = \exp\left([\boldsymbol{r}^{(k)}]_\times\right) \boldsymbol{R}^{(k)}, \qquad \boldsymbol{R}_i = \exp\left([\boldsymbol{r}_i^{(k)}]_\times\right) \boldsymbol{R}^{(k)}, \tag{C.5}$$

for $i = 1, 2$. When $\boldsymbol{r}^{(k)}$ is small, we can approximate $\boldsymbol{R}\boldsymbol{R}_i^{-1} = \exp([\boldsymbol{r}^{(k)}]_\times) \exp(-[\boldsymbol{r}_i^{(k)}]_\times)$ as $\exp([\boldsymbol{r}^{(k)} - \boldsymbol{r}_i^{(k)}]_\times)$. Using this approximation, (C.4) reduces to a quadratic optimization problem that we solve to obtain a better estimate $\boldsymbol{R}^*$:

$$\boldsymbol{r}^{(k+1)} = \min_{\boldsymbol{r}^{(k)}} \quad \frac{1}{2}(\boldsymbol{r}^{(k)} - \boldsymbol{r}_1^{(k)})^T \boldsymbol{C}_1^{-1} (\boldsymbol{r}^{(k)} - \boldsymbol{r_1}^{(k)}) + \\ \frac{1}{2}(\boldsymbol{r}^{(k)} - \boldsymbol{r}_2^{(k)})^T \boldsymbol{C}_2^{-1} (\boldsymbol{r}^{(k)} - \boldsymbol{r_2}^{(k)}), \tag{C.6}$$

$\boldsymbol{r}_i^{(k)}$ is defined by (C.5) and computed using $\log^\vee(\boldsymbol{R}_i \boldsymbol{R}^{(k)^{-1}})$. The solution of this quadratic optimization problem is:

$$\boldsymbol{r}^{(k+1)} = \boldsymbol{C}\left(\boldsymbol{C}_1^{-1}\boldsymbol{r}_1^{(k)} + \boldsymbol{C}_2^{-1}\boldsymbol{r}_2^{(k)}\right). \tag{C.7}$$

A better approximation $\boldsymbol{R}^{(k+1)}$ of the likelihood is then computed by:

$$\boldsymbol{R}^{(k+1)} = \exp\left([\boldsymbol{r}^{(k+1)}]_\times\right) \boldsymbol{R}^{(k)}, \tag{C.8}$$

after which the complete procedure can be repeated to further improve the estimate until the updates $r^{(k)}$ become sufficiently small.

The covariance matrix $\boldsymbol{C}$ on our estimates $\boldsymbol{r}^{(k+1)}$ is not influenced by the above iterations and is given by:

$$\boldsymbol{C}^{-1} = \boldsymbol{C}_1^{-1} + \boldsymbol{C}_2^{-1} \tag{C.9}$$

The complete procedure is found in Algorithm 4. The above algorithm was first proposed for quaternions and dual quaternions in [54] but was adapted here to rotation matrices. Our covariance matrices are also expressed in the tool or world frame instead of the object frame.

The algorithm results in a unique solution in a few iterations, unless there is a 180° difference between the orientations $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$. The latter cannot occur since the rotation matrices were aligned using our Algorithm 3.

Appendix D

Equivalent Stiffness and Damping

This appendix derives expressions for the values of the equivalent stiffness and damping constant that result from the control strategy introduced in Section 3.5 in the case of a 1-DOF system. The pose and wrench constraints (3.37) and (3.38) are adapted to that case as:

$$v_p = k_p(p_{des} - p_{act}) + v_{des}, \tag{D.1}$$

$$v_f = k_f \hat{C}_f(f_{des} - f_{act}) + v_{des}. \tag{D.2}$$

The constraint-based controller weights these constraints with w_p and w_f. Without loss of generality the weights are selected such that $w_f = 1 - w_p$, which results in a combined velocity:

$$v_{act} = w_p v_p + w_f v_f. \tag{D.3}$$

Substituting (D.1)-(D.2) in (D.3) yields:

$$\begin{aligned} v_{act} = w_p\big(k_p(p_{des} - p_{act}) + v_{des}\big) + \\ w_f\big(k_f \hat{C}_f(f_{des} - f_{act}) + v_{des}\big). \end{aligned} \tag{D.4}$$

Further elaborating (D.4), the equation of a spring-damper system can be found:

$$\frac{1}{w_f k_f \hat{C}_f}(v_{des} - v_{act}) + \frac{w_p k_p}{w_f k_f \hat{C}_f}(p_{des} - p_{act}) = f_{act} - f_{des}. \tag{D.5}$$

Accordingly, the equivalent stiffness and damping constant are:

$$k_{eq} = \frac{w_p k_p}{w_f k_f \hat{C}_f} \text{ and } b_{eq} = \frac{1}{w_f k_f \hat{C}_f}. \tag{D.6}$$

Appendix E

Relative Orientation of FS Frames Derived from Data Expressed in Two Viewpoints

The aim of this appendix is to find the Frenet-Serret FS moving frames, $[mf_1]$ and $[mf_2]$, in two viewpoints and their relation for a general vector $\boldsymbol{c}$. This vector can be replaced with rotational velocity $\boldsymbol{\omega}$, translational velocity $\boldsymbol{p}'$, force $\boldsymbol{f}$, or moment $\boldsymbol{m}$. It is important to note that the orientation of an ISA frame follows the same behavior as an FS frame. Therefore, the conclusions drawn in this appendix regarding the orientation of the FS frame also apply to the orientation of the ISA frame.

E.1 Vectors and their Derivatives in Two Viewpoints

This procedure requires expressing both the general vector $\boldsymbol{c}$ and its corresponding derivative $\boldsymbol{c}'$ in two viewpoints. We assume that a vector $\boldsymbol{c}$ is defined as follows in the first viewpoint represented by a reference frame

$[o_1]$.[1,2,3]

$$ {}^{o_2}_{o_1}\boldsymbol{c}^{o_2}_{o_1} \triangleq \boldsymbol{c} \tag{E.1} $$

We also assume that the rotation matrix between the two rigid bodies is known. Then, we can calculate $\boldsymbol{c}$ with respect to $[o_2]$ as follows:

$$ \begin{aligned} {}^{o_1}_{o_2}\boldsymbol{c}^{o_2}_{o_2} &= -{}^{o_2}_{o_1}\boldsymbol{R}_{o_1} \ {}^{o_2}_{o_1}\boldsymbol{c}^{o_2}_{o_1} \\ &= -{}^{o_1}_{o_2}\boldsymbol{R}_{o_2} \ \boldsymbol{c}. \end{aligned} \tag{E.2} $$

The derivative of $\boldsymbol{c}$ in $[o_1]$ is defined by differentiating (E.1):

$$ {}^{o_2}_{o_1}\boldsymbol{c}'^{o_2}_{o_1} = \boldsymbol{c}', \tag{E.3} $$

while the derivative of $\boldsymbol{c}$ in $[o_2]$ is found by differentiating (E.2):[1]

$$ \begin{aligned} {}^{o_1}_{o_2}\boldsymbol{c}'^{o_2}_{o_2} &= -\left({}^{o_1}_{o_2}\boldsymbol{R}'_{o_2}\boldsymbol{c} + {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{c}'\right) \\ &= -\left({}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\left[{}^{o_1}_{o_1}\boldsymbol{\omega}_{o_2}\right]_\times \boldsymbol{c} + {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{c}'\right) \\ &= -\left(-{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\left[\boldsymbol{\omega}\right]_\times \boldsymbol{c} + {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{c}'\right) \\ &= -(-{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}(\boldsymbol{\omega}\times\boldsymbol{c}) + {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{c}'). \end{aligned} \tag{E.4} $$

The vector $\boldsymbol{c}$ can be replaced by $\boldsymbol{\omega}$, $\boldsymbol{p}'$, $\boldsymbol{f}$, and $\boldsymbol{m}$. Note that if $\boldsymbol{c}$ is replaced by $\boldsymbol{\omega}$, $\boldsymbol{\omega}\times\boldsymbol{c}=\boldsymbol{0}$.

E.2 Unit Vectors of FS Frames in Two Viewpoints

E.2.1 Tangent Axis of FS Frames

The tangent axis of the moving frame $[mf_1]$ is defined with respect to $[o_1]$ as follows using (2.14) and (E.1).

$$ {}^{mf_1}_{o_1}\boldsymbol{e}_{t\,o_1} = \frac{\boldsymbol{c}}{\|\boldsymbol{c}\|} \tag{E.5} $$

[1] As explained in the List of Symbols, ${}^{b}_{c}\boldsymbol{c}^{d}_{a}$ is a 3×1 general vector representing a quantity associated with rigid body *"b"* with respect to rigid body *"a"* at reference point d and expressed in reference orientation frame $[c]$.

[2] In case of $\boldsymbol{\omega}$ and $\boldsymbol{f}$, the reference point is irrelevant, whereas for $\boldsymbol{p}'$ and $\boldsymbol{m}$, the reference point does matter. Therefore, we include the reference point without loss of generality.

[3] In the remainder of this appendix, if the leading/trailing subscript/superscript of the vector $\boldsymbol{c}$ are as shown in (E.1), the leading/trailing subscript/superscript will be omitted for simplicity.

[1] In the remainder of this appendix, if the leading/trailing subscript/superscript of the vector $\boldsymbol{p}$ are as shown in (E.3), the leading/trailing subscript/superscript will be omitted for simplicity.

The tangent axis of the moving frame $[mf_2]$ is likewise defined with respect to $[o_2]$ as follows using (2.14) and (E.2):

$$ {}^{mf_2}_{o_2}\boldsymbol{e}_{t\,o_2} = \frac{-{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{c}}{\left\|-{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{c}\right\|}, \tag{E.6} $$

or,

$$ {}^{mf_2}_{o_2}\boldsymbol{e}_{t\,o_2} = -{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\frac{\boldsymbol{c}}{\|\boldsymbol{c}\|}. \tag{E.7} $$

After inspecting both tangent axes, (E.5) and (E.7), we conclude that the following must hold, represented in $[o_1]$. Remark that the tangent axes are in opposite directions.

$$ {}^{mf_1}_{o_1}\boldsymbol{e}_{t\,o_1} = -{}^{mf_2}_{o_1}\boldsymbol{e}_{t\,o_1} \tag{E.8} $$

E.2.2 Normal Axis of FS Frames

The normal axis of the moving frame $[mf_1]$ is defined with respect to $[o_1]$ as follows using (2.14), (E.1), and (E.3).

$$ {}^{mf_1}_{o_1}\boldsymbol{e}_{n\,o_1} = \frac{(\boldsymbol{c}\times\boldsymbol{c}')\times\boldsymbol{c}}{\|(\boldsymbol{c}\times\boldsymbol{c}')\times\boldsymbol{c}\|} \tag{E.9} $$

The normal axis of the moving frame $[mf_2]$ is likewise defined with respect to $[o_2]$ as follows using (2.14), (E.2), and (E.4).

$$ \begin{aligned} {}^{mf_2}_{o_2}\boldsymbol{e}_{n\,o_2} &= \frac{\Big((-{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{c})\times-\big(-{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}(\boldsymbol{\omega}\times\boldsymbol{c})+{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{c}'\big)\Big)\times(-{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{c})}{\left\|\Big((-{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{c})\times-\big(-{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}(\boldsymbol{\omega}\times\boldsymbol{c})+{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{c}'\big)\Big)\times(-{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{c})\right\|} \\ &= -{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\frac{\big(\boldsymbol{c}\times(-\boldsymbol{\omega}\times\boldsymbol{c}+\boldsymbol{c}')\big)\times\boldsymbol{c}}{\left\|\big(\boldsymbol{c}\times(-\boldsymbol{\omega}\times\boldsymbol{c}+\boldsymbol{c}')\big)\times\boldsymbol{c}\right\|} \\ &= -{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\frac{\big((\boldsymbol{\omega}\times\boldsymbol{c})\times\boldsymbol{c}+\boldsymbol{c}\times\boldsymbol{c}')\big)\times\boldsymbol{c}}{\left\|\big((\boldsymbol{\omega}\times\boldsymbol{c})\times\boldsymbol{c}+\boldsymbol{c}\times\boldsymbol{c}')\big)\times\boldsymbol{c}\right\|} \end{aligned} \tag{E.10} $$

The following relation holds between the normal axes, (E.9) and (E.10), represented with respect to $[o_1]$.

$$ {}^{mf_2}_{o_1}\boldsymbol{e}_{n\,o_1} = -\frac{\big((\boldsymbol{\omega}\times\boldsymbol{c})\times\boldsymbol{c}\big)\times\boldsymbol{c}+\|(\boldsymbol{c}\times\boldsymbol{c}')\times\boldsymbol{c}\|\,{}^{mf_1}_{o_1}\boldsymbol{e}_{n\,o_1}}{\left\|\big((\boldsymbol{\omega}\times\boldsymbol{c})\times\boldsymbol{c}+\boldsymbol{c}\times\boldsymbol{c}')\big)\times\boldsymbol{c}\right\|} \tag{E.11} $$

Note that if $\boldsymbol{c}=\boldsymbol{\omega}$, then $\boldsymbol{\omega}\times\boldsymbol{c}=\boldsymbol{0}$, which simplifies (E.11).

E.2.3 Binormal Axis of FS Frames

The binormal axis of the moving frame $[mf_1]$ is defined with respect to $[o_1]$ as follows using (E.5) and (E.9) by cross product.

$$\begin{aligned} {}^{mf_1}_{o_1}\boldsymbol{e}_{b\,o_1} &= {}^{mf_1}_{o_1}\boldsymbol{e}_{t\,o_1} \times {}^{mf_1}_{o_1}\boldsymbol{e}_{n\,o_1} \\ &= \frac{\boldsymbol{c}}{\|\boldsymbol{c}\|} \times \frac{(\boldsymbol{c}\times\boldsymbol{c}')\times\boldsymbol{c}}{\|(\boldsymbol{c}\times\boldsymbol{c}')\times\boldsymbol{c}\|}, \end{aligned} \tag{E.12}$$

We use *vector triple product*, $\boldsymbol{a}\times(\boldsymbol{b}\times\boldsymbol{c}) = \boldsymbol{b}(\boldsymbol{a}\cdot\boldsymbol{c}) - \boldsymbol{c}(\boldsymbol{a}\cdot\boldsymbol{b})$, to rewrite ${}^{mf_1}_{o_1}\boldsymbol{e}_{n\,o_1}$ as follows:

$${}^{mf_1}_{o_1}\boldsymbol{e}_{n\,o_1} = \frac{\|\boldsymbol{c}\|^2\boldsymbol{c}' - (\boldsymbol{c}\cdot\boldsymbol{c}')\boldsymbol{c}}{\left\|\|\boldsymbol{c}\|^2\boldsymbol{c}' - (\boldsymbol{c}\cdot\boldsymbol{c}')\boldsymbol{c}\right\|}. \tag{E.13}$$

This results in the following equation.

$${}^{mf_1}_{o_1}\boldsymbol{e}_{b\,o_1} = \frac{\boldsymbol{c}}{\|\boldsymbol{c}\|} \times \frac{\|\boldsymbol{c}\|^2\boldsymbol{c}' - (\boldsymbol{c}\cdot\boldsymbol{c}')\boldsymbol{c}}{\left\|\|\boldsymbol{c}\|^2\boldsymbol{c}' - (\boldsymbol{c}\cdot\boldsymbol{c}')\boldsymbol{c}\right\|} \tag{E.14}$$

Since $\boldsymbol{c}\times\boldsymbol{c} = \boldsymbol{0}$, the relation is simplified as follows:

$${}^{mf_1}_{o_1}\boldsymbol{e}_{b\,o_1} = \frac{\boldsymbol{c}\times\boldsymbol{c}'}{\|\boldsymbol{c}\times\boldsymbol{c}'\|}. \tag{E.15}$$

Similarly, the binormal axis of the moving frame $[mf_2]$ is likewise defined with respect to $[o_2]$ as follows using (E.2) and (E.4),

$${}^{mf_2}_{o_2}\boldsymbol{e}_{b\,o_2} = \frac{\left(-{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{c}\right)\times\left(-\left(-{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}(\boldsymbol{\omega}\times\boldsymbol{c}) + {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{c}'\right)\right)}{\left\|\left(-{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{c}\right)\times\left(-\left(-{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}(\boldsymbol{\omega}\times\boldsymbol{c}) + {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{c}'\right)\right)\right\|}, \tag{E.16}$$

or in another form as follows:

$${}^{mf_2}_{o_2}\boldsymbol{e}_{b\,o_2} = {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\frac{\left((\boldsymbol{\omega}\times\boldsymbol{c})\times\boldsymbol{c} + \boldsymbol{c}\times\boldsymbol{c}'\right)}{\left\|\left((\boldsymbol{\omega}\times\boldsymbol{c})\times\boldsymbol{c} + \boldsymbol{c}\times\boldsymbol{c}'\right)\right\|}. \tag{E.17}$$

The following relation holds between the binormal axes, (E.15) and (E.17), represented with respect to $[o_1]$.

$${}^{mf_2}_{o_1}\boldsymbol{e}_{b\,o_1} = \frac{\left((\boldsymbol{\omega}\times\boldsymbol{c})\times\boldsymbol{c} + \|\boldsymbol{c}\times\boldsymbol{c}'\|\,{}^{mf_1}_{o_1}\boldsymbol{e}_{b\,o_1}\right)}{\left\|\left((\boldsymbol{\omega}\times\boldsymbol{c})\times\boldsymbol{c} + \boldsymbol{c}\times\boldsymbol{c}'\right)\right\|} \tag{E.18}$$

Note that if $\boldsymbol{c} = \boldsymbol{\omega}$, then $\boldsymbol{\omega}\times\boldsymbol{c} = \boldsymbol{0}$, which simplifies (E.18).

E.3 Rotation Matrix between Two FS Frames

The angle between ${}^{mf_1}_{o_1}\boldsymbol{e}_{n\,o_1}$ and ${}^{mf_2}_{o_1}\boldsymbol{e}_{n\,o_1}$, named ϕ_c, can be found by the following equation.

$$\phi_c = \arccos\left({}^{mf_1}_{o_1}\boldsymbol{e}_{n\,o_1} \cdot {}^{mf_2}_{o_1}\boldsymbol{e}_{n\,o_1}\right) \tag{E.19}$$

Now, we can construct the following rotation matrix which expresses $[mf_2]$ with respect to $[mf_1]$ considering (E.8), (E.11), and (E.18):

$${}^{mf_2}_{mf_1}\boldsymbol{R}_{mf_1} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -\mathrm{c}\phi_c & \mathrm{s}\phi_c \\ 0 & \mathrm{s}\phi_c & \mathrm{c}\phi_c \end{pmatrix}, \tag{E.20}$$

where $\mathrm{c}\phi_c$ and $\mathrm{s}\phi_c$ stand for $\cos(\phi_c)$ and $\sin(\phi_c)$, respectively. This rotation matrix is visualized in Figure 5.2a.

E.4 Special Case: No Rotation between Rigid Bodies

From the relation between the normal axes in equation (E.11), we can conclude that if $\boldsymbol{\omega} = \mathbf{0}$, the normal axes are aligned but point in opposite directions. Similarly, based on the relation between the binormal axes in equation (E.18), we observe that when $\boldsymbol{\omega} = \mathbf{0}$, the binormal axes are fully aligned in both direction and orientation. This implies that the relation between the orientations of the FS frames is simplified if there is no rotation between the rigid bodies.

E.5 Special Case: Relative Orientation of FS Frames of Rotation

In the special case of $\boldsymbol{c} = \boldsymbol{\omega}$, the condition $\boldsymbol{c} \times \boldsymbol{\omega} = \mathbf{0}$ implies that the normal axes of the FS frames align but are oppositely oriented, as shown in (E.11). Similarly, the binormal axes of the FS frames fully coincide, as illustrated in (E.18). This configuration is depicted in Figure 5.2b. This explanation results in the following relation between the FS frames derived from rotational velocities in two viewpoints.

$${}^{rot_2}_{rot_1}\boldsymbol{R}_{rot_1} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{E.21}$$

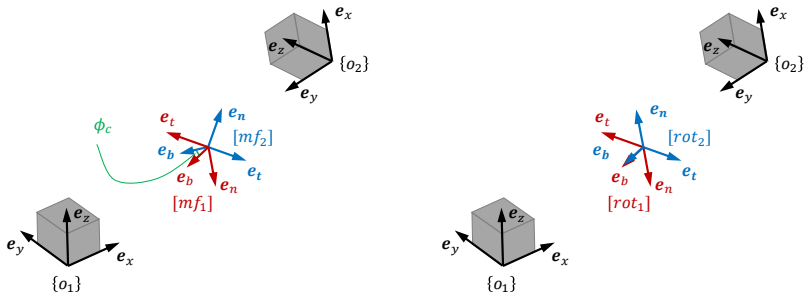


Figure 5.2: FS frames calculated in two viewpoints for: (a) a general case based on ${}^{o2}_{o1}\boldsymbol{c}^{o2}_{o1}$ and ${}^{o1}_{o2}\boldsymbol{c}^{o2}_{o2}$ expressed in $[o_1]$, (b) the special case of rotation based on ${}^{o2}_{o1}\boldsymbol{\omega}_{o_1}$ and ${}^{o1}_{o2}\boldsymbol{\omega}_{o_2}$ expressed in $[o_1]$.

Appendix F

Relative Position of ISA Frames Derived from Data Expressed in Two Viewpoints

The aim of this appendix is to find the ISA moving frames, $\{mf_1\}$ and $\{mf_2\}$, in two viewpoints and their relation for a general screw $\boldsymbol{\mathfrak{z}}$. This screw can be replaced with twist $\boldsymbol{t}$ or wrench $\boldsymbol{w}$.

F.1 Screws and their Derivatives in Two Viewpoints

To derive the relations between the moving frames, we begin with expressing a screw in the first reference frame $\{o_1\}$ and the second reference frame $\{o_2\}$. We also need the derivative of this screw in the aforementioned reference frames. Then, we are able to derive the relations between the moving frames, $\{mf_1\}$ and $\{mf_2\}$. We assume that a screw $\boldsymbol{\mathfrak{z}}$, which can be twist $\boldsymbol{t}$ or wrench $\boldsymbol{w}$, is

defined as follows:[1,2]

$$ {}^{o_2}_{o_1}\boldsymbol{\jmath}^{o_1}_{o_1} = \begin{pmatrix} {}^{o_2}_{o_1}\boldsymbol{a}_{o_1} \\ {}^{o_2}_{o_1}\boldsymbol{b}^{o_1}_{o_1} \end{pmatrix} \triangleq \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{pmatrix}. \tag{F.1} $$

This screw can be expressed with respect to $\{o_2\}$ using the screw transformation matrix ${}^{o_1}\boldsymbol{S}_{o_2}$.[1]

$$ \begin{aligned} {}^{o_1}_{o_2}\boldsymbol{\jmath}^{o_2}_{o_2} &= -\,{}^{o_1}\boldsymbol{S}_{o_2}\ {}^{o_2}_{o_1}\boldsymbol{\jmath}^{o_1}_{o_1} \\ &= -\begin{pmatrix} {}^{o_1}_{o_2}\boldsymbol{R}_{o_2} & \boldsymbol{a} \\ {}^{o_1}_{o_2}\boldsymbol{R}_{o_2} & \boldsymbol{b} - {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\left[{}^{o_2}_{o_1}\boldsymbol{p}_{o_1}\right]_\times \boldsymbol{a} \end{pmatrix} \\ &= -\begin{pmatrix} {}^{o_1}_{o_2}\boldsymbol{R}_{o_2} & \boldsymbol{a} \\ {}^{o_1}_{o_2}\boldsymbol{R}_{o_2} & \boldsymbol{b} - {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\left[\boldsymbol{p}\right]_\times \boldsymbol{a} \end{pmatrix} \end{aligned} \tag{F.2} $$

The derivative of the screw in $\{o_1\}$ is defined as follows using (F.1),

$$ {}^{o_2}_{o_1}\boldsymbol{\jmath}'^{o_1}_{o_1} = \begin{pmatrix} \boldsymbol{a}' \\ \boldsymbol{b}' \end{pmatrix}, \tag{F.3} $$

[1] As explained in the List of Symbols, ${}^b_c\boldsymbol{\jmath}^d_a$ is a 6×1 general screw representing a quantity associated with rigid body *"b"* with respect to rigid body *"a"* at reference point d and expressed in reference orientation frame $[c]$. This screw consists of two 3×1 vectors known as directional component ${}^b_c\boldsymbol{a}_a$ and moment component ${}^b_c\boldsymbol{b}^d_a$.

[2] In the remainder of this appendix, if the leading/trailing subscript/superscript of the direction and moment parts of the screw ($\boldsymbol{f}$ and $\boldsymbol{m}$) are as shown in (F.1), the leading/trailing subscript/superscript will be omitted for simplicity.

[1] In the remainder of this appendix, if the leading/trailing subscript/superscript of the vector $\boldsymbol{p}$ are as shown in (F.2), the leading/trailing subscript/superscript will be omitted for simplicity.

while the derivative of the screw in $\{o_2\}$ is found by differentiating (F.2).

$$
\begin{aligned}
&{}^{o_1}_{o_2}\boldsymbol{\jmath}'^{o_2}_{o_2} = \\
&-\begin{pmatrix} {}^{o_1}_{o_2}\boldsymbol{R}'_{o_2}\boldsymbol{a} + {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{a}' \\ {}^{o_1}_{o_2}\boldsymbol{R}'_{o_2}\boldsymbol{b} + {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{b}' - {}^{o_1}_{o_2}\boldsymbol{R}'_{o_2}(\boldsymbol{p}\times\boldsymbol{a}) - {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}(\boldsymbol{p}'\times\boldsymbol{a}) - {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}(\boldsymbol{p}\times\boldsymbol{a}') \end{pmatrix} = \\
&-\begin{pmatrix} {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\left[{}^{o_1}_{o_1}\boldsymbol{\omega}_{o_2}\right]_\times \boldsymbol{a} + {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{a}' \\ {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\left(\left[{}^{o_1}_{o_1}\boldsymbol{\omega}_{o_2}\right]_\times \boldsymbol{b} + \boldsymbol{b}' - \left[{}^{o_1}_{o_1}\boldsymbol{\omega}_{o_2}\right]_\times (\boldsymbol{p}\times\boldsymbol{a}) - \boldsymbol{p}'\times\boldsymbol{a} - \boldsymbol{p}\times\boldsymbol{a}' \right) \end{pmatrix} = \\
&-\begin{pmatrix} -{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\left[\boldsymbol{\omega}\right]_\times \boldsymbol{a} + {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{a}' \\ {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\left(-\left[\boldsymbol{\omega}\right]_\times \boldsymbol{b} + \boldsymbol{b}' + \left[\boldsymbol{\omega}\right]_\times (\boldsymbol{p}\times\boldsymbol{a}) - \boldsymbol{p}'\times\boldsymbol{a} - \boldsymbol{p}\times\boldsymbol{a}' \right) \end{pmatrix}
\end{aligned}
\tag{F.4}
$$

Note that if $\boldsymbol{\jmath} = \boldsymbol{\ell}$, then $\boldsymbol{a} = \boldsymbol{\omega}$ which results in $[\boldsymbol{\omega}]_\times \boldsymbol{a} = \boldsymbol{0}$.

F.2 Position of ISA Frames in Two Viewpoints

In Appendix E, it is shown that the tangent axes of the FS moving frames derived from two viewpoints are oriented in opposite directions, while the remaining two axes may differ by an angle $\phi_\jmath$. This behavior also applies to the orientation of the ISA moving frames. Building on this insight, we propose the schematic in Figure 5.4. We further assume that the origins of the moving frames do not necessarily coincide; their spatial separation is represented by the position vector ${}^{mf_2}_{mf_1}\boldsymbol{p}_{mf_1}$. However, based on the findings of Appendix E, this separation is restricted to lie along the ISA, since the tangent axes must remain in opposite directions.

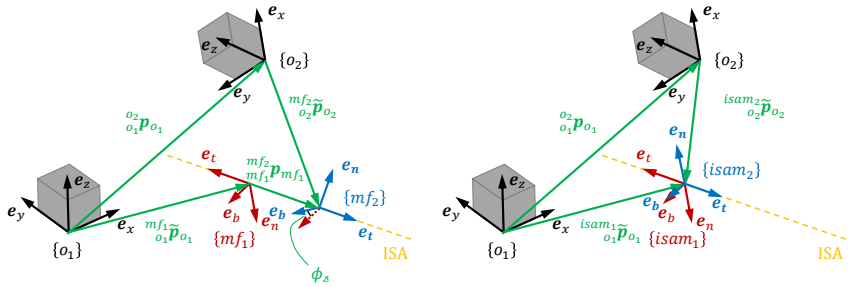


Figure 5.4: ISA frames calculated in two viewpoints for: (a) a general case based on $^{o_2}_{o_1}\boldsymbol{\jmath}^{o_1}_{o_1}$ and $^{o_1}_{o_2}\boldsymbol{\jmath}^{o_2}_{o_2}$ expressed in $\{o_1\}$ (b) the special case of motion based on $^{o_2}_{o_1}\boldsymbol{t}^{o_1}_{o_1}$ and $^{o_1}_{o_2}\boldsymbol{t}^{o_2}_{o_2}$ expressed in $\{o_1\}$.

Accordingly, this appendix investigates the validity of the relation illustrated as a loop with green arrows in Figure 5.4a. This relation is expressed here with respect to the first reference frame, $\{o_1\}$.

$$^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}_{o_1} + {}^{mf_1}_{o_1}\widetilde{\boldsymbol{R}}_{o_1}\ {}^{mf_2}_{mf_1}\boldsymbol{p}_{mf_1} = {}^{o_2}_{o_1}\boldsymbol{p}_{o_1} + {}^{o_2}_{o_1}\boldsymbol{R}_{o_1}\ {}^{mf_2}_{o_2}\widetilde{\boldsymbol{p}}_{o_2} \tag{F.5}$$

Since it has shown in Appendix E that the tangent vectors of the ISA frames are oriented in opposite directions, we can conclude the following relation for $^{mf_2}_{mf_1}\boldsymbol{p}_{mf_1}$.

$$^{mf_2}_{mf_1}\boldsymbol{p}_{mf_1} = \begin{pmatrix} p_t & 0 & 0 \end{pmatrix}^T \tag{F.6}$$

We continue with finding $^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}_{o_1}$ and $^{mf_2}_{o_2}\widetilde{\boldsymbol{p}}_{o_2}$ in the next two subsections.

F.2.1 Position of the First ISA Frame in the First Viewpoint

We aim to find $^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}_{o_1}$ which is the first term in (F.5) and formulated as follows using (2.35).

$$^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}_{o_1} = {}^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}_{\perp o_1} + {}^{mf_1}\widetilde{p}_{\parallel o_1}\ {}^{mf_1}_{o_1}\boldsymbol{e}_{t\,o_1} \tag{F.7}$$

The first term in the right-hand side of (F.7), $^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}_{\perp o_1}$, is as follows according to (2.33),

$$^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}_{\perp o_1} = \frac{\boldsymbol{a} \times \boldsymbol{b}}{\|\boldsymbol{a}\|^2}, \tag{F.8}$$

and its derivative is as follows:

$$ {}^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}'_{\perp o_1} = \frac{\left(\boldsymbol{a}' \times \boldsymbol{b} + \boldsymbol{a} \times \boldsymbol{b}'\right) \|\boldsymbol{a}\|^2 - 2\left(\boldsymbol{a} \times \boldsymbol{b}\right)\left(\boldsymbol{a}.\boldsymbol{a}'\right)}{\|\boldsymbol{a}\|^4}. \tag{F.9} $$

We assign letters to some of its terms for ease of manipulation as follows:

$$ {}^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}'_{\perp o_1} = \frac{(A+B)\|\boldsymbol{a}\|^2 - 2CD}{\|\boldsymbol{a}\|^4}, \tag{F.10} $$

where,

$$ \begin{aligned} A_1 &= \boldsymbol{a}' \times \boldsymbol{b} \\ B_1 &= \boldsymbol{a} \times \boldsymbol{b}' \\ C_1 &= \boldsymbol{a} \times \boldsymbol{b} \\ D_1 &= \boldsymbol{a} \cdot \boldsymbol{a}'. \end{aligned} \tag{F.11} $$

The second term in the right-hand side of (F.7), ${}^{mf_1}\widetilde{p}_{\parallel o_1}$, is as follows according to (2.34).

$$ {}^{mf_1}\widetilde{p}_{\parallel o_1} = \frac{{}^{mf_1}_{o_1}\boldsymbol{e}_{n\,o_1} \cdot {}^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}'_{\perp o_1}}{\omega_\kappa^{mf_1}} \tag{F.12} $$

The normal unit vector ${}^{mf_1}_{o_1}\boldsymbol{e}_{n\,o_1}$ and ${}^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}'_{\perp o_1}$ can be defined by (E.9) and (F.9), respectively. The invariant descriptor ω_κ is also defined by (2.19) which requires (F.1) and (F.3),

$$ \omega_\kappa^{mf_1} = \frac{\|\boldsymbol{a} \times \boldsymbol{a}'\|}{\|\boldsymbol{a}\|^2} \tag{F.13} $$

The last term in the right-hand side of (F.7), ${}^{mf_1}_{o_1}\boldsymbol{e}_{t\,o_1}$, is addressed in (E.5). Now, we can rewrite (F.7) by substituting (F.8), (F.12), and (E.5) in it.

$$
\begin{aligned}
{}^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}_{o_1} &= {}^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}_{\perp o_1} + {}^{mf_1}\widetilde{p}_{\parallel o_1}\,{}^{mf_1}_{o_1}\boldsymbol{e}_{t\,o_1} \\
&= {}^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}_{\perp o_1} + \frac{{}^{mf_1}_{o_1}\boldsymbol{e}_{n\,o_1} \cdot {}^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}'_{\perp o_1}}{\omega_\kappa^{mf_1}}\;\; {}^{mf_1}_{o_1}\boldsymbol{e}_{t\,o_1} \\
&= \frac{\boldsymbol{a}\times\boldsymbol{b}}{\|\boldsymbol{a}\|^2} \\
&+ \frac{\left(\frac{(\boldsymbol{a}\times\boldsymbol{a}')\times\boldsymbol{a}}{\|(\boldsymbol{a}\times\boldsymbol{a}')\times\boldsymbol{a}\|}\right)\cdot\left(\frac{\left(\boldsymbol{a}'\times\boldsymbol{b}+\boldsymbol{a}\times\boldsymbol{b}'\right)\|\boldsymbol{a}\|^2-2(\boldsymbol{a}\times\boldsymbol{b})\left(\boldsymbol{a}\cdot\boldsymbol{a}'\right)}{\|\boldsymbol{a}\|^4}\right)}{\frac{\|\boldsymbol{a}\times\boldsymbol{a}'\|}{\|\boldsymbol{a}\|^2}}\left(\frac{\boldsymbol{a}}{\|\boldsymbol{a}\|}\right) \\
&= \frac{\boldsymbol{a}\times\boldsymbol{b}}{\|\boldsymbol{a}\|^2} \\
&+ \frac{\big((\boldsymbol{a}\times\boldsymbol{a}')\times\boldsymbol{a}\big)\cdot\big((\boldsymbol{a}'\times\boldsymbol{b}+\boldsymbol{a}\times\boldsymbol{b}')\|\boldsymbol{a}\|^2-2(\boldsymbol{a}\times\boldsymbol{b})(\boldsymbol{a}\cdot\boldsymbol{a}')\big)}{\|\boldsymbol{a}\times\boldsymbol{a}'\|\,\|\boldsymbol{a}\|^4}\boldsymbol{a}
\end{aligned}
\tag{F.14}
$$

F.2.2 Position of the Second ISA Frame in the Second Viewpoint

Similar to the previous subsection, we aim to find ${}^{mf_2}_{o_2}\widetilde{\boldsymbol{p}}_{o_2}$ which is the last term in (F.5) and formulated as follows using (2.35).

$$
{}^{mf_2}_{o_2}\widetilde{\boldsymbol{p}}_{o_2} = {}^{mf_2}_{o_2}\widetilde{\boldsymbol{p}}_{\perp o_2} + {}^{mf_2}\widetilde{p}_{\parallel o_2}\,{}^{mf_2}_{o_2}\boldsymbol{e}_{t\,o_2} \tag{F.15}
$$

The first term in the right-hand side of (F.15), ${}^{mf_2}_{o_2}\widetilde{\boldsymbol{p}}_{\perp o_2}$, is as follows according to (2.33),

$$
\begin{aligned}
{}^{mf_2}_{o_2}\widetilde{\boldsymbol{p}}_{\perp o_2} &= \frac{{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\;\boldsymbol{a}\times\left({}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\;\boldsymbol{b}-{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\;[\boldsymbol{p}]_\times\,\boldsymbol{a}\right)}{\left\|-{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\;\boldsymbol{a}\right\|^2} \\
&= \frac{{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\;\left(\boldsymbol{a}\times\boldsymbol{b}-\boldsymbol{a}\times(\boldsymbol{p}\times\boldsymbol{a})\right)}{\|\boldsymbol{a}\|^2}
\end{aligned}
\tag{F.16}
$$

and its derivative is as follows which is in a similar format as (F.11).

$$
\begin{aligned}
A_2 &= \big(- {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\ (\boldsymbol{\omega}\times\boldsymbol{a}) + {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\ \boldsymbol{a}'\big) \times \big({}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{b} - {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\ (\boldsymbol{p}\times\boldsymbol{a})\big) \\
&= {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\big(\boldsymbol{b}\times(\boldsymbol{\omega}\times\boldsymbol{a}) + (\boldsymbol{\omega}\times\boldsymbol{a})\times(\boldsymbol{p}\times\boldsymbol{a}) + \boldsymbol{a}'\times\boldsymbol{b} \\
&\quad - \boldsymbol{a}'\times(\boldsymbol{p}\times\boldsymbol{a})\big) \\
B_2 &= {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{a}\times\Big({}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\big(-\boldsymbol{\omega}\times\boldsymbol{b} + \boldsymbol{b}' + \boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{a}) \\
&\quad - \boldsymbol{p}'\times\boldsymbol{a} - \boldsymbol{p}\times\boldsymbol{a}'\big)\Big) \\
&= {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\Big(\boldsymbol{a}\times(\boldsymbol{b}\times\boldsymbol{\omega}) + \boldsymbol{a}\times\boldsymbol{b}' + \boldsymbol{a}\times\big(\boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{a})\big) \\
&\quad - \boldsymbol{a}\times(\boldsymbol{p}'\times\boldsymbol{a}) - \boldsymbol{a}\times(\boldsymbol{p}\times\boldsymbol{a}')\Big) \\
C_2 &= {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{a}\times\big({}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{b} - {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}({}^{o_1}_{o_1}\boldsymbol{p}_{o_2}\times\boldsymbol{a})\big) \\
&= {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\ (\boldsymbol{a}\times\boldsymbol{b} - \boldsymbol{a}\times(\boldsymbol{p}\times\boldsymbol{a})) \\
D_2 &= {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{a}\cdot\big(-{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}(\boldsymbol{\omega}\times\boldsymbol{a}) + {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\boldsymbol{a}'\big) \\
&= -\boldsymbol{a}\cdot(\boldsymbol{\omega}\times\boldsymbol{a}) + \boldsymbol{a}\cdot\boldsymbol{a}' \\
&= \boldsymbol{a}\cdot\boldsymbol{a}'
\end{aligned}
\tag{F.17}
$$

This results in the following equation.

$$
\begin{aligned}
{}^{mf_2}_{o_2}\widetilde{\boldsymbol{p}}'_{\perp o_2} = \frac{{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}}{\|\boldsymbol{a}\|^4}\Big(&\big(\boldsymbol{b}\times(\boldsymbol{\omega}\times\boldsymbol{a}) + (\boldsymbol{\omega}\times\boldsymbol{a})\times(\boldsymbol{p}\times\boldsymbol{a}) + \boldsymbol{a}'\times\boldsymbol{b} \\
&- \boldsymbol{a}'\times(\boldsymbol{p}\times\boldsymbol{a})\big)\|\boldsymbol{a}\|^2 \\
&+ \Big(\boldsymbol{a}\times(\boldsymbol{b}\times\boldsymbol{\omega}) + \boldsymbol{a}\times\boldsymbol{b}' + \boldsymbol{a}\times\big(\boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{a})\big) \\
&- \boldsymbol{a}\times(\boldsymbol{p}'\times\boldsymbol{a}) - \boldsymbol{a}\times(\boldsymbol{p}\times\boldsymbol{a}')\Big)\|\boldsymbol{a}\|^2 \\
&- 2(\boldsymbol{a}\cdot\boldsymbol{a}')\big(\boldsymbol{a}\times\boldsymbol{b} - \boldsymbol{a}\times(\boldsymbol{p}\times\boldsymbol{a})\big)\Big)
\end{aligned}
\tag{F.18}
$$

The second term in the right-hand side of (F.15), ${}^{mf_2}\widetilde{p}_{\parallel o_2}$, is as follows according to (2.34).

$$ {}^{mf_2}\widetilde{p}_{\parallel o_2} = \frac{{}^{mf_2}_{o_2}\boldsymbol{e}_{n\,o_2} \cdot {}^{mf_2}_{o_2}\widetilde{\boldsymbol{p}}'_{\perp o_2}}{\omega_\kappa^{mf_2}} \tag{F.19} $$

The normal unit vector ${}^{mf_2}_{o_2}\boldsymbol{e}_{n\,o_2}$ and ${}^{mf_2}_{o_2}\widetilde{\boldsymbol{p}}'_{\perp o_2}$ can be defined by (E.10) and (F.18), respectively. The invariant descriptor ω_κ is also defined by (2.19) which requires (F.2) and (F.4).

$$ \begin{aligned} \omega_\kappa^{mf_2} &= \frac{\left\| {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\ \boldsymbol{a} \times {}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\ \left(-(\boldsymbol{\omega}\times\boldsymbol{a}) + \boldsymbol{a}'\right)\right\|}{\left\| -{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\ \boldsymbol{a}\right\|^2} \\ &= \frac{\|(\boldsymbol{\omega}\times\boldsymbol{a})\times\boldsymbol{a} + \boldsymbol{a}\times\boldsymbol{a}'\|}{\|\boldsymbol{a}\|^2} \end{aligned} \tag{F.20} $$

After substituting the known components, we obtain the following relation.

$$ \begin{aligned} {}^{mf_2}\widetilde{p}_{\parallel o_2} &= \frac{{}^{mf_2}_{o_2}\boldsymbol{e}_{n\,o_2} \cdot {}^{mf_2}_{o_2}\widetilde{\boldsymbol{p}}'_{\perp o_2}}{\omega_\kappa^{mf_2}} \\ &= \begin{pmatrix} -{}^{o_1}_{o_2}\boldsymbol{R}_{o_2} \dfrac{\left((\boldsymbol{\omega}\times\boldsymbol{a})\times\boldsymbol{a}+\boldsymbol{a}\times\boldsymbol{a}'\right)\times\boldsymbol{a}}{\left\|\left((\boldsymbol{\omega}\times\boldsymbol{a})\times\boldsymbol{a}+\boldsymbol{a}\times\boldsymbol{a}'\right)\times\boldsymbol{a}\right\|} \\ \dfrac{\|(\boldsymbol{\omega}\times\boldsymbol{a})\times\boldsymbol{a}+\boldsymbol{a}\times\boldsymbol{a}'\|}{\|\boldsymbol{a}\|^2} \end{pmatrix} \cdot \\ &\quad \frac{{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}}{\|\boldsymbol{a}\|^4}\Big(\big(\boldsymbol{b}\times(\boldsymbol{\omega}\times\boldsymbol{a}) + (\boldsymbol{\omega}\times\boldsymbol{a})\times(\boldsymbol{p}\times\boldsymbol{a}) + \boldsymbol{a}'\times\boldsymbol{b} \\ &\quad - \boldsymbol{a}'\times(\boldsymbol{p}\times\boldsymbol{a})\big)\|\boldsymbol{a}\|^2 \\ &\quad + \Big(\boldsymbol{a}\times(\boldsymbol{b}\times\boldsymbol{\omega}) + \boldsymbol{a}\times\boldsymbol{b}' + \boldsymbol{a}\times\big(\boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{a})\big) \\ &\quad - \boldsymbol{a}\times(\boldsymbol{p}'\times\boldsymbol{a}) - \boldsymbol{a}\times(\boldsymbol{p}\times\boldsymbol{a}')\Big)\|\boldsymbol{a}\|^2 \\ &\quad - 2(\boldsymbol{a}\cdot\boldsymbol{a}')\big(\boldsymbol{a}\times\boldsymbol{b} - \boldsymbol{a}\times(\boldsymbol{p}\times\boldsymbol{a})\big)\Big) \end{aligned} \tag{F.21} $$

This can be further rewritten as follows:

$$
\begin{aligned}
{}^{mf_2}\widetilde{p}_{\|\,o_2} = \Bigg(& \frac{-\big((\boldsymbol{\omega}\times\boldsymbol{a})\times\boldsymbol{a}+\boldsymbol{a}\times\boldsymbol{a}'\big)\times\boldsymbol{a}\,\|\boldsymbol{a}\|^2}{\|(\boldsymbol{\omega}\times\boldsymbol{a})\times\boldsymbol{a}+\boldsymbol{a}\times\boldsymbol{a}'\|^2\,\|\boldsymbol{a}\|}\Bigg)\cdot \\
& \frac{1}{\|\boldsymbol{a}\|^4}\Big(\big(\boldsymbol{b}\times(\boldsymbol{\omega}\times\boldsymbol{a})+(\boldsymbol{\omega}\times\boldsymbol{a})\times(\boldsymbol{p}\times\boldsymbol{a})+\boldsymbol{a}'\times\boldsymbol{b} \\
& -\boldsymbol{a}'\times(\boldsymbol{p}\times\boldsymbol{a})\big)\|\boldsymbol{a}\|^2 \\
& +\big(\boldsymbol{a}\times(\boldsymbol{b}\times\boldsymbol{\omega})+\boldsymbol{a}\times\boldsymbol{b}'+\boldsymbol{a}\times\big(\boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{a})\big) \\
& -\boldsymbol{a}\times(\boldsymbol{p}'\times\boldsymbol{a})-\boldsymbol{a}\times(\boldsymbol{p}\times\boldsymbol{a}')\big)\|\boldsymbol{a}\|^2 \\
& -2(\boldsymbol{a}\cdot\boldsymbol{a}')\big(\boldsymbol{a}\times\boldsymbol{b}-\boldsymbol{a}\times(\boldsymbol{p}\times\boldsymbol{a})\big)\Big). \qquad \text{(F.22)}
\end{aligned}
$$

The last term in the right-hand side of (F.15), ${}^{mf_2}_{o_2}\boldsymbol{e}_{t\,o_2}$, is addressed in (E.7). By substituting (F.16), (F.22), and (E.7) into (F.15), we can rewrite it and further simplify the expression by canceling out several terms. From this point on, we use strike-through notation for canceled terms in the equations to facilitate easier tracking of the simplifications.

$$
\begin{aligned}
{}^{mf_2}_{o_2}\widetilde{\boldsymbol{p}}_{o_2} &= {}^{mf_2}_{o_2}\widetilde{\boldsymbol{p}}_{\perp o_2} + {}^{mf_2}\widetilde{p}_{\|\,o_2}\,{}^{mf_2}_{o_2}\boldsymbol{e}_{t\,o_2} \\
&= \frac{{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\big(\boldsymbol{a}\times\boldsymbol{b}-\boldsymbol{a}\times(\boldsymbol{p}\times\boldsymbol{a})\big)}{\|\boldsymbol{a}\|^2}+ \\
&\quad \Bigg(\frac{\cancel{\big((\boldsymbol{\omega}\times\boldsymbol{a})\times\boldsymbol{a}+\boldsymbol{a}\times\boldsymbol{a}'\big)\times\boldsymbol{a}\|\boldsymbol{a}\|}}{\|(\boldsymbol{\omega}\times\boldsymbol{a})\times\boldsymbol{a}+\boldsymbol{a}\times\boldsymbol{a}'\|^2}\Bigg)\cdot \\
&\quad \frac{1}{\|\boldsymbol{a}\|^4}\Big(\big(\boldsymbol{b}\times(\boldsymbol{\omega}\times\boldsymbol{a})+(\boldsymbol{\omega}\times\boldsymbol{a})\times(\boldsymbol{p}\times\boldsymbol{a})+\boldsymbol{a}'\times\boldsymbol{b} \\
&\quad -\boldsymbol{a}'\times(\boldsymbol{p}\times\boldsymbol{a})\big)\|\boldsymbol{a}\|^2 \\
&\quad +\big(\boldsymbol{a}\times(\boldsymbol{b}\times\boldsymbol{\omega})+\boldsymbol{a}\times\boldsymbol{b}'+\boldsymbol{a}\times\big(\boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{a})\big) \\
&\quad -\boldsymbol{a}\times(\boldsymbol{p}'\times\boldsymbol{a})-\boldsymbol{a}\times(\boldsymbol{p}\times\boldsymbol{a}')\big)\|\boldsymbol{a}\|^2 \\
&\quad -2(\boldsymbol{a}\cdot\boldsymbol{a}')\big(\boldsymbol{a}\times\boldsymbol{b}-\boldsymbol{a}\times(\boldsymbol{p}\times\boldsymbol{a})\big)\Big)\Bigg(\frac{\cancel{{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}\,\boldsymbol{a}}}{\cancel{\|\boldsymbol{a}\|}}\Bigg) \qquad \text{(F.23)}
\end{aligned}
$$

F.3 Transformation Matrix between Two ISA Frames

The positions of the two moving frames, ${}^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}_{o_1}$ and ${}^{mf_2}_{o_2}\widetilde{\boldsymbol{p}}_{o_2}$, are determined using (F.14) and (F.23). Since the remaining terms in (F.5) are already known, we can determine ${}^{mf_2}_{mf_1}\boldsymbol{p}_{mf_1}$.

$${}^{mf_1}_{o_1}\widetilde{\boldsymbol{p}}_{o_1} + {}^{mf_1}_{o_1}\widetilde{\boldsymbol{R}}_{o_1}\ {}^{mf_2}_{mf_1}\boldsymbol{p}_{mf_1} = {}^{o_2}_{o_1}\boldsymbol{p}_{o_1} + {}^{o_2}_{o_1}\boldsymbol{R}_{o_1}\ {}^{mf_2}_{o_2}\widetilde{\boldsymbol{p}}_{o_2} \tag{F.5}$$

With this information, we can construct a transformation matrix that describes the relation between the ISA frames. The orientation between the ISA frames is the same as the orientation of the FS frames derived in Appendix E and provided in (E.20). Thus, the rotation part of the transformation matrix is already known.

$${}^{mf_2}_{mf_1}\boldsymbol{R}_{mf_1} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -\mathrm{c}\phi_c & \mathrm{s}\phi_c \\ 0 & \mathrm{s}\phi_c & \mathrm{c}\phi_c \end{pmatrix} \tag{E.20}$$

Note that ϕ_c should be replaced by $\phi_\jmath$. The relation between the positions of the ISA frames has already been shown as follows:

$${}^{mf_2}_{mf_1}\boldsymbol{p}_{mf_1} = \begin{pmatrix} p_t & 0 & 0 \end{pmatrix}^T. \tag{F.6}$$

Using (E.20) and (F.6) we can construct the transformation matrix ${}^{mf_2}_{mf_1}\boldsymbol{T}_{mf_1}$ as follows:

$${}^{mf_2}_{mf_1}\boldsymbol{T}_{mf_1} = \begin{pmatrix} -1 & 0 & 0 & p_t \\ 0 & -\mathrm{c}\phi_\jmath & \mathrm{s}\phi_\jmath & 0 \\ 0 & \mathrm{s}\phi_\jmath & \mathrm{c}\phi_\jmath & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{F.24}$$

F.4 Special Case: No Rotation and/or No Translation between Rigid Bodies

The relative position of the ISA frames was derived in the previous section for a general screw $\boldsymbol{\jmath}$. In this section, we focus on two special cases that simplify the motion between the moving frames.

F.4.1 No Rotation between Rigid Bodies

In the case where there is no relative rotation between the two rigid bodies ($\boldsymbol{\omega} = \boldsymbol{0}$), but translation is present, the orientations of the ISA frames follow

those of the FS frames, discussed in Section E.4. However, the positions of the ISA frames may still differ, since the relation described by (F.23) does not reduce to that of (F.14). This indicates that, even without rotational motion, translational effects can result in different frame origins for the ISA representations.

F.4.2 No Rotation and No Translation between Rigid Bodies

When there is neither relative rotation nor translation between the two rigid bodies ($\boldsymbol{\omega} = 0$ and $\boldsymbol{p} = 0$), then (F.23) simplifies to (F.14). This implies that, the origins of the two ISA frames coincide. Note that the orientations of the ISA frames still follow the orientation of FS frames, discussed in Section E.4.

F.5 Special Case: Relative Position of ISA Frames of Motion

To analyze the relative position of the motion ISA frames derived from two viewpoints, (F.5) simplifies. Assuming that the origins of the two moving frames coincide, we derive the following equation. In this subsection, we provide a proof that this relation holds for this special case.

$$ {}^{isam_1}_{o_1}\widetilde{\boldsymbol{p}}_{o_1} = {}^{o_2}_{o_1}\boldsymbol{p}_{o_1} + {}^{o_2}_{o_1}\boldsymbol{R}_{o_1}\, {}^{isam_2}_{o_2}\widetilde{\boldsymbol{p}}_{o_2} \tag{F.25} $$

The derived expressions (F.14) and (F.23) are substituted into (F.25), with the terms adapted for motion. We then proceed to simplify the resulting equation.

$$
\begin{aligned}
&\frac{\boldsymbol{\omega}\times\boldsymbol{v}}{\|\boldsymbol{\omega}\|^2}+\\
&\frac{\big((\boldsymbol{\omega}\times\boldsymbol{\omega}')\times\boldsymbol{\omega}\big)\cdot\big(\,(\boldsymbol{\omega}'\times\boldsymbol{v}+\boldsymbol{\omega}\times\boldsymbol{v}')\,\|\boldsymbol{\omega}\|^2-2\,(\boldsymbol{\omega}\times\boldsymbol{v})\,(\boldsymbol{\omega}\cdot\boldsymbol{\omega}')\,\big)}{\|\boldsymbol{\omega}\times\boldsymbol{\omega}'\|^2\,\|\boldsymbol{\omega}\|^4}\boldsymbol{\omega}\\
&=\boldsymbol{p}+\cancel{{}^{o_2}_{o_1}\boldsymbol{R}_{o_1}}\left(\frac{\cancel{{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}}\big(\boldsymbol{\omega}\times\boldsymbol{v}-\boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{\omega})\,\big)}{\|\boldsymbol{\omega}\|^2}+\left(\frac{(\boldsymbol{\omega}\times\boldsymbol{\omega}')\times\boldsymbol{\omega}\cancel{\|\boldsymbol{\omega}\|}}{\|\boldsymbol{\omega}\times\boldsymbol{\omega}'\|^2}\right)\cdot\right.\\
&\left.\frac{1}{\|\boldsymbol{\omega}\|^4}\Big(\big(\boldsymbol{\omega}'\times\boldsymbol{v}-\boldsymbol{\omega}'\times(\boldsymbol{p}\times\boldsymbol{\omega})\big)\|\boldsymbol{\omega}\|^2\right.\\
&+\big(\boldsymbol{\omega}\times\boldsymbol{v}'-\boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{\omega}')\,\big)\|\boldsymbol{\omega}\|^2\\
&\left.-2(\boldsymbol{\omega}\cdot\boldsymbol{\omega}')\big(\boldsymbol{\omega}\times\boldsymbol{v}-\boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{\omega})\big)\Big)\left(\frac{\cancel{{}^{o_1}_{o_2}\boldsymbol{R}_{o_2}}\boldsymbol{\omega}}{\cancel{\|\boldsymbol{\omega}\|}}\right)\right)
\end{aligned}
\tag{F.26}
$$

We continue by performing further cancellations on both sides of the equation.

$$
\begin{aligned}
&\frac{\cancel{\boldsymbol{\omega}\times\boldsymbol{v}}}{\|\boldsymbol{\omega}\|^2}+\\
&\frac{\big((\boldsymbol{\omega}\times\boldsymbol{\omega}')\times\boldsymbol{\omega}\big)\cdot\big(\,\cancel{(\boldsymbol{\omega}'\times\boldsymbol{v}+\boldsymbol{\omega}\times\boldsymbol{v}')}\,\|\boldsymbol{\omega}\|^2-\cancel{2\,(\boldsymbol{\omega}\times\boldsymbol{v})\,(\boldsymbol{\omega}\cdot\boldsymbol{\omega}')}\,\big)}{\|\boldsymbol{\omega}\times\boldsymbol{\omega}'\|^2\,\|\boldsymbol{\omega}\|^4}\boldsymbol{\omega}\\
&=\boldsymbol{p}+\frac{\big(\cancel{\boldsymbol{\omega}\times\boldsymbol{v}}-\boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{\omega})\,\big)}{\|\boldsymbol{\omega}\|^2}+\Big(\frac{(\boldsymbol{\omega}\times\boldsymbol{\omega}')\times\boldsymbol{\omega}}{\|\boldsymbol{\omega}\times\boldsymbol{\omega}'\|^2}\Big)\cdot\\
&\frac{1}{\|\boldsymbol{\omega}\|^4}\Big(\big(\cancel{\boldsymbol{\omega}'\times\boldsymbol{v}}-\boldsymbol{\omega}'\times(\boldsymbol{p}\times\boldsymbol{\omega})\big)\|\boldsymbol{\omega}\|^2\\
&+\big(\cancel{\boldsymbol{\omega}\times\boldsymbol{v}'}-\boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{\omega}')\,\big)\|\boldsymbol{\omega}\|^2\\
&-2(\boldsymbol{\omega}\cdot\boldsymbol{\omega}')\big(\cancel{\boldsymbol{\omega}\times\boldsymbol{v}}-\boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{\omega})\big)\Big)\boldsymbol{\omega}
\end{aligned}
\tag{F.27}
$$

Equation (F.27) is further simplified by canceling out similar terms from both sides of the equality. Using *vector triple product*, $\boldsymbol{a}\times(\boldsymbol{b}\times\boldsymbol{c})=\boldsymbol{b}(\boldsymbol{a}\cdot\boldsymbol{c})-\boldsymbol{c}(\boldsymbol{a}\cdot\boldsymbol{b})$, we can eliminate $\boldsymbol{p}$. Furthermore, $\boldsymbol{\omega}$ is also canceled out in the process as shown

in the following.

$$
\begin{aligned}
0 = \cancel{\boldsymbol{p}} - \frac{\cancel{\boldsymbol{p}}(\boldsymbol{\omega}\cdot\boldsymbol{\omega}) - \big(\cancel{\boldsymbol{\omega}}(\boldsymbol{\omega}\cdot\boldsymbol{p})\big)}{\|\boldsymbol{\omega}\|^2} + \Big(\frac{(\boldsymbol{\omega}\times\boldsymbol{\omega}')\times\boldsymbol{\omega}}{\|\boldsymbol{\omega}\times\boldsymbol{\omega}'\|^2\,\|\boldsymbol{\omega}\|^4}\Big)\cdot \\
&\Big(\big(-\boldsymbol{\omega}'\times(\boldsymbol{p}\times\boldsymbol{\omega})\big)\|\boldsymbol{\omega}\|^2 \\
&+\big(-\boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{\omega}')\big)\|\boldsymbol{\omega}\|^2 \\
&-2(\boldsymbol{\omega}\cdot\boldsymbol{\omega}')\big(-\boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{\omega})\big)\Big)\cancel{\boldsymbol{\omega}}
\end{aligned}
\tag{F.28}
$$

This results in the following equation.

$$
\begin{aligned}
0 = \frac{\boldsymbol{\omega}\cdot\boldsymbol{p}}{\|\boldsymbol{\omega}\|^2} + \Big(\frac{(\boldsymbol{\omega}\times\boldsymbol{\omega}')\times\boldsymbol{\omega}}{\|\boldsymbol{\omega}\times\boldsymbol{\omega}'\|^2\,\|\boldsymbol{\omega}\|^4}\Big)\cdot \\
&\Big(\big(-\boldsymbol{\omega}'\times(\boldsymbol{p}\times\boldsymbol{\omega})\big)\|\boldsymbol{\omega}\|^2 \\
&+\big(-\boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{\omega}')\big)\|\boldsymbol{\omega}\|^2 \\
&-2(\boldsymbol{\omega}\cdot\boldsymbol{\omega}')\big(-\boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{\omega})\big)\Big)
\end{aligned}
\tag{F.29}
$$

To simplify the following equation further, we assign letters to some of its terms for clarity and ease of manipulation.

$$
\begin{aligned}
0 = \frac{\boldsymbol{\omega}\cdot\boldsymbol{p}}{\|\boldsymbol{\omega}\|^2} + \frac{\overbrace{\big((\boldsymbol{\omega}\times\boldsymbol{\omega}')\times\boldsymbol{\omega}\big)\cdot\Big(\big(-\boldsymbol{\omega}'\times(\boldsymbol{p}\times\boldsymbol{\omega})\big)\|\boldsymbol{\omega}\|^2\Big)}^{E}}{\|\boldsymbol{\omega}\times\boldsymbol{\omega}'\|^2\,\|\boldsymbol{\omega}\|^4} \\
+ \frac{\overbrace{\big((\boldsymbol{\omega}\times\boldsymbol{\omega}')\times\boldsymbol{\omega}\big)\cdot\Big(\big(-\boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{\omega}')\big)\|\boldsymbol{\omega}\|^2\Big)}^{F}}{\|\boldsymbol{\omega}\times\boldsymbol{\omega}'\|^2\,\|\boldsymbol{\omega}\|^4} \\
+ \frac{\overbrace{\big((\boldsymbol{\omega}\times\boldsymbol{\omega}')\times\boldsymbol{\omega}\big)\cdot\Big(-2(\boldsymbol{\omega}\cdot\boldsymbol{\omega}')\big(-\boldsymbol{\omega}\times(\boldsymbol{p}\times\boldsymbol{\omega})\big)\Big)}^{G}}{\|\boldsymbol{\omega}\times\boldsymbol{\omega}'\|^2\,\|\boldsymbol{\omega}\|^4}
\end{aligned}
\tag{F.30}
$$

Using vector triple product, $\boldsymbol{a} \times (\boldsymbol{b} \times \boldsymbol{c}) = \boldsymbol{b}(\boldsymbol{a} \cdot \boldsymbol{c}) - \boldsymbol{c}(\boldsymbol{a} \cdot \boldsymbol{b})$, we can calculate the following terms from equation (F.30).

$$\begin{aligned} E &= \big(\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{\omega}')\big) \cdot \Big(\big(-\boldsymbol{\omega}' \times (\boldsymbol{p} \times \boldsymbol{\omega})\big) \|\boldsymbol{\omega}\|^2\Big) \\ &= -\big(\boldsymbol{\omega}(\boldsymbol{\omega} \cdot \boldsymbol{\omega}') - \boldsymbol{\omega}'(\boldsymbol{\omega} \cdot \boldsymbol{\omega})\big) \cdot \big(\boldsymbol{p}(\boldsymbol{\omega}' \cdot \boldsymbol{\omega}) - \boldsymbol{\omega}(\boldsymbol{\omega}' \cdot \boldsymbol{p})\big) \|\boldsymbol{\omega}\|^2 \\ &= -(\boldsymbol{\omega} \cdot \boldsymbol{p}) \|\boldsymbol{\omega} \cdot \boldsymbol{\omega}'\|^2 \|\boldsymbol{\omega}\|^2 + (\boldsymbol{\omega} \cdot \boldsymbol{\omega}')(\boldsymbol{\omega}' \cdot \boldsymbol{p}) \|\boldsymbol{\omega}\|^4 + \\ &(\boldsymbol{\omega}' \cdot \boldsymbol{p})(\boldsymbol{\omega}' \cdot \boldsymbol{\omega}) \|\boldsymbol{\omega}\|^4 - (\boldsymbol{\omega}' \cdot \boldsymbol{\omega})(\boldsymbol{\omega}' \cdot \boldsymbol{p}) \|\boldsymbol{\omega}\|^4 \end{aligned} \tag{F.31}$$

$$\begin{aligned} F &= \big(\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{\omega}')\big) \cdot \Big(\big(-\boldsymbol{\omega} \times (\boldsymbol{p} \times \boldsymbol{\omega}')\big) \|\boldsymbol{\omega}\|^2\Big) \\ &= -\big(\boldsymbol{\omega}(\boldsymbol{\omega} \cdot \boldsymbol{\omega}') - \boldsymbol{\omega}'(\boldsymbol{\omega} \cdot \boldsymbol{\omega})\big) \cdot \big(\boldsymbol{p}(\boldsymbol{\omega}' \cdot \boldsymbol{\omega}) - \boldsymbol{\omega}'(\boldsymbol{p} \cdot \boldsymbol{\omega})\big) \|\boldsymbol{\omega}\|^2 \\ &= -(\boldsymbol{\omega} \cdot \boldsymbol{p}) \|\boldsymbol{\omega} \cdot \boldsymbol{\omega}'\|^2 \|\boldsymbol{\omega}\|^2 + \|\boldsymbol{\omega} \cdot \boldsymbol{\omega}'\|^2 (\boldsymbol{p} \cdot \boldsymbol{\omega}) \|\boldsymbol{\omega}\|^2 + \\ &(\boldsymbol{\omega}' \cdot \boldsymbol{p})(\boldsymbol{\omega}' \cdot \boldsymbol{\omega}) \|\boldsymbol{\omega}\|^4 - \|\boldsymbol{\omega}'\|^2 (\boldsymbol{p} \cdot \boldsymbol{\omega}) \|\boldsymbol{\omega}\|^4 \end{aligned} \tag{F.32}$$

$$\begin{aligned} G &= \big(\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{\omega}')\big) \cdot \Big(-2(\boldsymbol{\omega} \cdot \boldsymbol{\omega}')\big(-\boldsymbol{\omega} \times (\boldsymbol{p} \times \boldsymbol{\omega})\big)\Big) \\ &= \big(\boldsymbol{\omega}(\boldsymbol{\omega} \cdot \boldsymbol{\omega}') - \boldsymbol{\omega}'(\boldsymbol{\omega} \cdot \boldsymbol{\omega})\big) \cdot \big(\boldsymbol{p}(\boldsymbol{\omega} \cdot \boldsymbol{\omega}) - \boldsymbol{\omega}(\boldsymbol{\omega} \cdot \boldsymbol{p})\big) 2(\boldsymbol{\omega} \cdot \boldsymbol{\omega}') \\ &= 2(\boldsymbol{\omega} \cdot \boldsymbol{p}) \|\boldsymbol{\omega} \cdot \boldsymbol{\omega}'\|^2 \|\boldsymbol{\omega}\|^2 - 2 \|\boldsymbol{\omega}\|^2 \|\boldsymbol{\omega} \cdot \boldsymbol{\omega}'\|^2 (\boldsymbol{\omega} \cdot \boldsymbol{p}) \\ &- 2(\boldsymbol{\omega}' \cdot \boldsymbol{p}) \|\boldsymbol{\omega}\|^4 (\boldsymbol{\omega} \cdot \boldsymbol{\omega}') + 2 \|\boldsymbol{\omega}' \cdot \boldsymbol{\omega}\|^2 \|\boldsymbol{\omega}\|^2 (\boldsymbol{\omega} \cdot \boldsymbol{p}) \end{aligned} \tag{F.33}$$

Now, final results of (F.31)-(F.33) are written in (F.30) and many terms are canceled out.

$$0 = \frac{\boldsymbol{\omega} \cdot \boldsymbol{p}}{\|\boldsymbol{\omega}\|^2} + \left(\frac{\|\boldsymbol{\omega} \cdot \boldsymbol{\omega}'\|^2 (\boldsymbol{p} \cdot \boldsymbol{\omega}) \|\boldsymbol{\omega}\|^2 - \|\boldsymbol{\omega}'\|^2 (\boldsymbol{p} \cdot \boldsymbol{\omega}) \|\boldsymbol{\omega}\|^4}{\|\boldsymbol{\omega} \times \boldsymbol{\omega}'\|^2 \|\boldsymbol{\omega}\|^4}\right) \tag{F.34}$$

The following equation, $\|\boldsymbol{a} \times \boldsymbol{b}\|^2 + \|\boldsymbol{a} \cdot \boldsymbol{b}\|^2 = \|\boldsymbol{a}\|^2 \|\boldsymbol{b}\|^2$, commonly known as *Lagrange's identity*, is used to find the equivalent of $\|\boldsymbol{\omega} \cdot \boldsymbol{\omega}'\|^2$ as follows:

$$\|\boldsymbol{\omega} \cdot \boldsymbol{\omega}'\|^2 = \|\boldsymbol{\omega}\|^2 \|\boldsymbol{\omega}'\|^2 - \|\boldsymbol{\omega} \times \boldsymbol{\omega}'\|^2 . \tag{F.35}$$

By substituting (F.35) in (F.34), we achieve the following relation.

$$0 = \frac{\boldsymbol{\omega} \cdot \boldsymbol{p}}{\|\boldsymbol{\omega}\|^2} + \left(\frac{-\|\boldsymbol{\omega} \times \boldsymbol{\omega}'\|^2 (\boldsymbol{p} \cdot \boldsymbol{\omega}) \|\boldsymbol{\omega}\|^2}{\|\boldsymbol{\omega} \times \boldsymbol{\omega}'\|^2 \|\boldsymbol{\omega}\|^4} \right) \tag{F.36}$$

This equation can be simplified to the following equality.

$$0 = 0 \qquad \square \tag{F.37}$$

Therefore, it is proved that the following equation holds for origins of ISA frames.

$$\overset{isam_1}{o_1}\widetilde{\boldsymbol{p}}_{o_1} = \overset{o_2}{o_1}\boldsymbol{p}_{o_1} + \overset{o_2}{o_1}\boldsymbol{R}_{o_1} \overset{isam_2}{o_2}\widetilde{\boldsymbol{p}}_{o_2} \tag{F.25}$$

Thus, we can construct the following transformation matrix to explain the relation between the ISA frames. The rotation matrix has already been derived in Appendix E, as shown in (E.21).

$$\overset{isam_2}{isam_1}\boldsymbol{T}_{isam_1} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{F.38}$$

Appendix G

Validation of Relations between Invariant Descriptors and Moving Frames through Synthetic and Experimental Data

The aim of this appendix is to validate the mathematical relations derived in Chapter 5. This will be done in two scenarios via synthetic and experimental data, discussed in Sections G.1 and G.2, respectively.

G.1 Validation through Synthetic Data

G.1.1 Vector Invariants

To avoid repetition, this section does not include the validation of the relations between the vector invariants, as the directional component of a screw is a vector. Instead, the relations between the screw invariants across the two viewpoints will be validated in the next section.

G.1.2 Screw Invariants

Input Data and Ground Truth Invariants

To validate the findings from the previous sections, we conduct a numerical case study utilizing synthetic data. To mitigate the effects of the desired tolerances introduced in the OCPs (2.25) and (2.48), we apply discretized formulas to calculate the pose, twist, wrench, screw invariants, and moving frames in two different viewpoints. We also begin with non-singular synthetic screw invariants for both motion and wrench, as illustrated in Figures G.1 and G.2, to avoid the problems of singularities. Finally, we investigate the relations between the invariants derived from data expressed in two viewpoints.

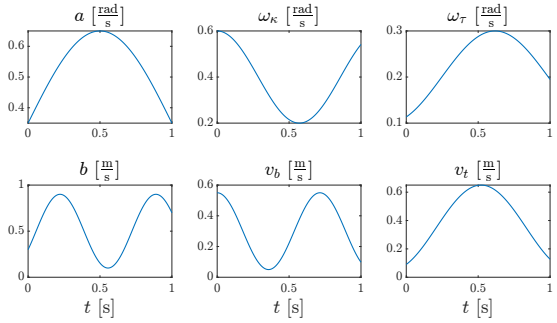


Figure G.1: Ground truth screw invariants of motion.

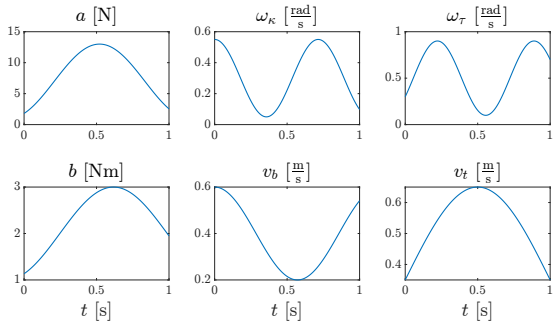


Figure G.2: Ground truth screw invariants of wrench.

We assume that the ground truth invariants are defined in the first viewpoint. Using these invariants, we can calculate pose, twist, and wrench trajectories in

this viewpoint. By applying an appropriate matrix transformation, we can also determine pose, twist, and wrench trajectories in the second viewpoint. The results for pose, twist, and wrench trajectories in both viewpoints are visualized in Figures G.3, G.4, and G.5, respectively.

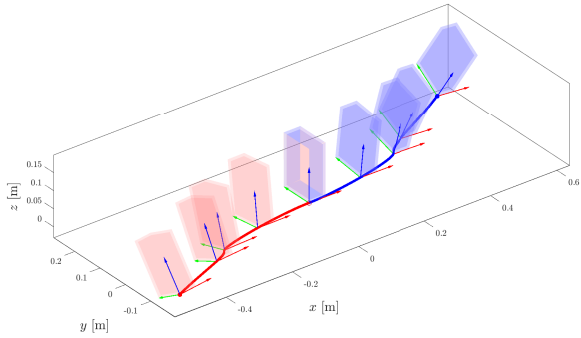


Figure G.3: Relative motion of the two rigid bodies. Blue: motion of $\{o_2\}$ with respect to $\{o_1\}$. Red: motion of $\{o_1\}$ with respect to $\{o_2\}$.

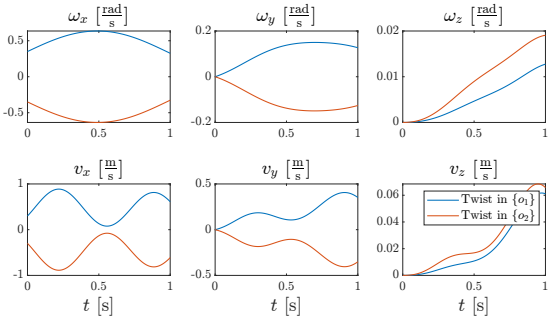


Figure G.4: Relative twist between the two rigid bodies. Blue: twist with respect to $\{o_1\}$. Red: twist with respect to $\{o_2\}$.

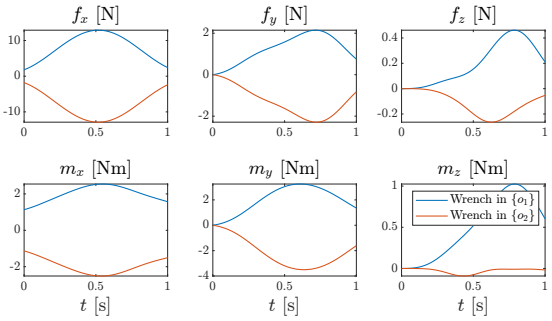


Figure G.5: Interaction wrench between the two rigid bodies. Blue: wrench applied to $\{o_1\}$ expressed in $\{o_1\}$. Red: wrench applied to $\{o_2\}$ expressed in $\{o_2\}$.

To calculate screw invariants we follow the following steps. First, using ground truth invariants of motion ${}_{isam_1}{}^{o_2}_{o_1}\widetilde{\boldsymbol{\jmath}}^{isam_1} t$ and ${}^{isam_1}_{isam_1}\boldsymbol{\imath}^{isam_1}_{o_1}$, we can reconstruct the object pose trajectory ${}^{o_2}_{o_1}\boldsymbol{T}_{o_1}$ and object twist ${}^{o_2}_{o_1}\boldsymbol{t}^{o_1}_{o_1}$. This is where *integration* occurs and we have to provide initial values for the moving frame $\{isam_2\}$ and object frame $\{o_2\}$ to construct the pose trajectory. In this study we selected identity matrices for these initial values. Furthermore, the object twist is transformed to the second viewpoint $\{o_2\}$ by screw transformation matrix,

$$ {}^{o_1}_{o_2}\boldsymbol{t}^{o_2}_{o_2} = -\boldsymbol{S}({}^{o_2}_{o_1}\boldsymbol{T}^{-1}_{o_1})\,{}^{o_2}_{o_1}\boldsymbol{t}^{o_1}_{o_1}. \tag{G.1} $$

Thus, we can calculate screw invariants and ISA frames in two viewpoints from twist data, ${}^{o_2}_{o_1}\boldsymbol{t}^{o_1}_{o_1}$ and ${}^{o_1}_{o_2}\boldsymbol{t}^{o_2}_{o_2}$. The results are shown in Figures G.3 and G.4 for pose and twist trajectories.

Similarly, the interaction wrench ${}^{o_2}_{o_1}\boldsymbol{w}^{o_1}_{o_1}$ can be reconstructed using the ground truth wrench invariants $({}_{isaw_1}{}^{o_2}_{o_1}\widetilde{\boldsymbol{\jmath}}^{isaw_1} w$ and ${}^{isaw_1}_{isaw_1}\boldsymbol{\imath}^{isaw_1}_{o_1})$. It is then transformed appropriately to the second viewpoint $\{o_2\}$, following the same approach as in (G.1), to obtain ${}^{o_1}_{o_2}\boldsymbol{w}^{o_2}_{o_2}$. The results are presented in Figure G.5.

Third, we calculate screw invariants of motion and wrench using discretized formulas. This results in ISA frames and invariants in two viewpoints. The results for screw invariants of motion and wrench are reported in the following two subsections.

Motion Invariants

Unsurprisingly, the invariants derived from twist data expressed in the first viewpoint correspond directly to the ground truth invariants, as illustrated in

Figure G.6. In contrast, the third and last invariants derived from the twist data expressed in the second viewpoint differ from the ground truth invariants.

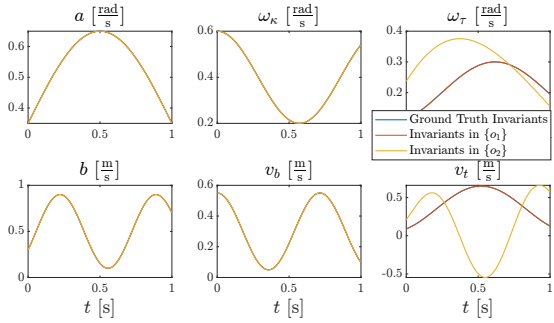
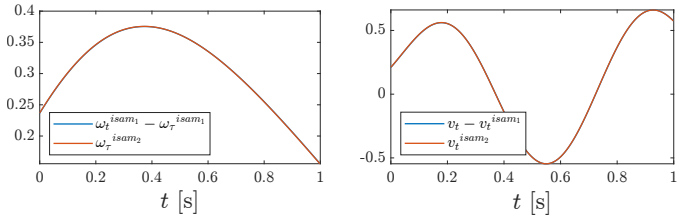


Figure G.6: Screw invariants of motion calculated from twist data expressed in two viewpoints.

To clarify the relations between the invariants ω_τ and v_t, we depict Figure G.7 to verify (5.34) and (5.36).



(a) (b)

Figure G.7: Relation between the invariants of motion. (a) Relation between ω_τ calculated from twist data expressed in two viewpoints. (b) Relation between v_t calculated from twist data expressed in two viewpoints.

To analyze this relation more accurately, we focus on the difference between the two moving frames. The distance between $\{isam_1\}$ and $\{isam_2\}$ can be calculated as follows:

$$\begin{pmatrix} 0 & -\delta\theta_b & \delta\theta_n & \delta d_t \\ \delta\theta_b & 0 & -\delta\theta_t & \delta d_n \\ -\delta\theta_n & \delta\theta_t & 0 & \delta d_b \\ 0 & 0 & 0 & 1 \end{pmatrix} = \log({\boldsymbol{T}^*}_{isam_1}^{isam_2} \boldsymbol{T}_{isam_1}) \tag{G.2}$$

where

$$\boldsymbol{T}^* = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{G.3}$$

is a transformation matrix to align the two moving frames. This describes a rotation of π about the binormal axis. With this premultiplication, we expect both ISA frames of motion to be fully aligned. The result is illustrated in Figure G.8, confirming the relation between the moving frames derived in (F.38).

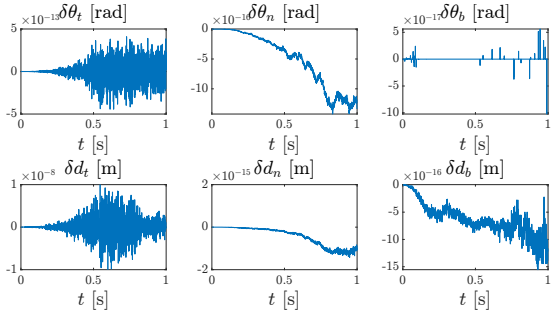


Figure G.8: Rotational (upper row) and translational distance (lower row) between the two ISA frames of motion calculated by (G.2) from twist data expressed in two viewpoints. The difference is in the machine precision, meaning the two frames coincide, confirming the configuration presented in Figure 5.4b.

To investigate the loop closure described by (5.23) for motion, we plot Figure G.9 to visualize ${}^{isam_2}_{isam_1}\boldsymbol{t}^{isam_1}_{isam_1}$. Unlike Figure G.8, we do not observe zero, particularly for the ω_t and v_t. This issue results from integration errors introduced by the discretized formulas explained in Appendices H and I.

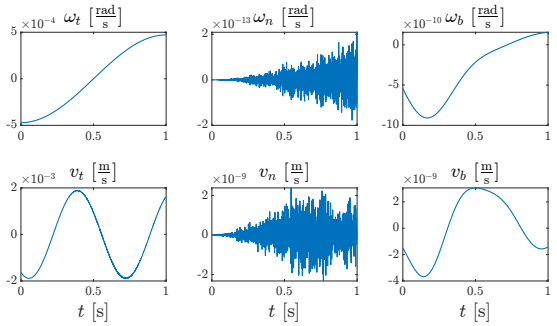


Figure G.9: Investigation of loop closure described in (5.23) for motion analysis, ${}^{isam_2}_{isam_1}\boldsymbol{t}^{isam_1}_{isam_1}$.

Wrench Invariants

Similar to the motion analysis, the wrench invariants derived from data expressed in the two viewpoints are calculated. In this analysis, differences in the invariants are observed across all four moving frame invariants, as illustrated in Figure G.10.

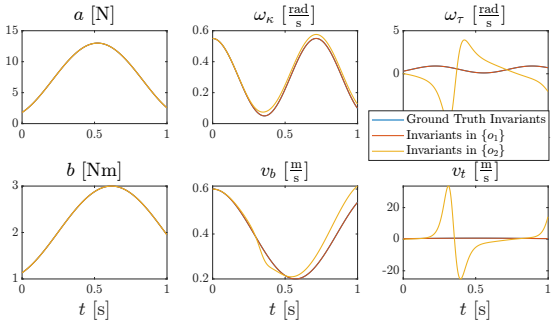


Figure G.10: Screw invariants of wrench calculated from wrench data expressed in two viewpoints.

We conduct a distance analysis between the moving frames, similar to what we performed in (G.2). The results are shown in Figure G.11. It can be concluded that the difference consists of the distance between the origins of the moving frames along the screw axis, δd_t, and the angle between the tangent axes of the frames, $\delta\theta_t$.

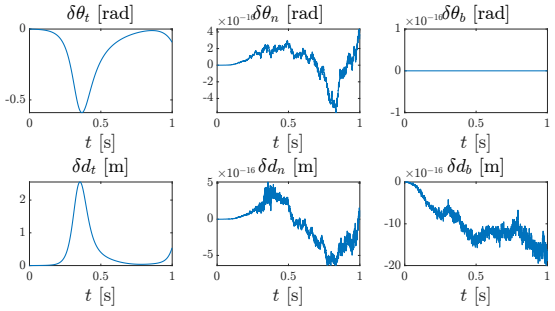


Figure G.11: Distance between the two ISA frames of wrench calculated by (G.2) from wrench data expressed in two viewpoints. The difference is in the machine precision except for the $\delta\theta_t$ and δd_t, confirming the configuration shown in Figure 5.4a.

Similar to motion analysis, the loop closure is analyzed in Figure G.12 for wrench analysis. Unlike Figure G.9, it is concluded that the difference between the moving frames involves a rotation around the tangent axis, ω_t, and a translation along the tangent axis v_t. However, the values of ω_n, ω_b, v_n, and v_b are not zero. This outcome, similar to that observed in motion analysis (as shown in Figure G.9), is attributed to integration errors in the discretized formulas explained in Appendices H and I.

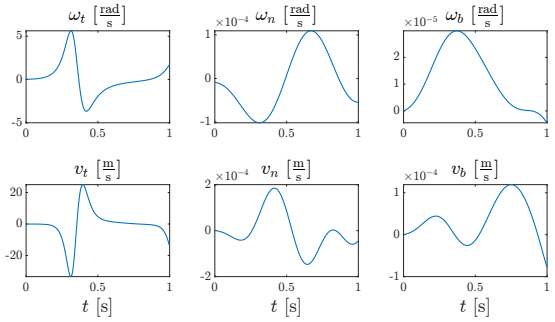
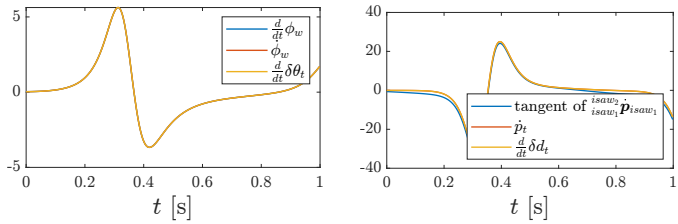


Figure G.12: Investigation of loop closure described in (5.23) for wrench analysis, ${}^{isaw_2}_{isaw_1}\boldsymbol{t}^{isaw_1}_{isaw_1}$.

To more clearly investigate the two sets of invariants and compare different equations, we present Figure G.13. In Figure G.13a, the first component $\frac{d}{dt}\phi_w$ is derived from (E.19), the second component $\dot{\phi}_w$ comes from (5.25), and the third component $\frac{d}{dt}\delta\theta_t$ is obtained by differentiating $\delta\theta_t$ in (G.2). In Figure G.13b,

the first component results from differentiating the tangential component of ${}^{isaw_2}_{isaw_1}\boldsymbol{p}_{isaw_1}$ in (F.5), the second component $\dot{p}_t$ comes from (5.25), and the third component $\frac{d}{dt}\delta d_t$ is obtained by differentiating δd_t in (G.2). This observation confirms the relation between the moving frames at twist level shown in (5.28) and (5.31).



(a) (b)

Figure G.13: The angle and distance between the two moving frames are calculated in different ways. (a) For the angle: the first component is the derivative of (E.19), the second component is achieved from (5.25), and the last component is found from differentiating (G.2). (b) For the distance analysis, the first component is the tangent component of the derivative of ${}^{isaw_1}\boldsymbol{R}^T_{o_1}\ {}^{isaw_2}_{o_1}\boldsymbol{p}_{isaw_1}$ found through (F.5), the second component is achieved from (5.25), and the last component is found from differentiating (G.2).

G.2 Validation through Real Experimental Data

As discussed in Section G.1.1, the validation of the relations between the vector invariants is not addressed here to avoid repetition, as the directional component of a screw is a vector. We utilize data from Section 2.5.5, specifically focusing on the calculated screw invariants with respect to the tracker (second viewpoint) and the world (first viewpoint), as illustrated in Figures 2.7a and 2.8. Thus, the invariants are calculated independently in the two viewpoints through OCPs. The second viewpoint has the second type of (near-)singularity because the direction of force remains almost unchanged during the demonstration. Additionally, we also investigate the motion invariants in the two viewpoints. The screw invariants of motion and wrench for each viewpoint are shown in Figures G.14 and G.15, respectively. Similar to the synthetic data, the object invariants are consistent across both viewpoints. For the moving frame invariants of motion, ω_κ and v_b are also comparable between the viewpoints, whereas the moving frame invariants of wrench exhibit noticeable differences. These findings are consistent with those observed in the synthetic data.

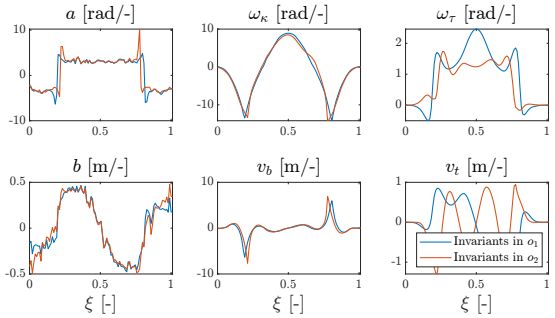


Figure G.14: Screw invariants of motion derived from pose data expressed in two viewpoints. Although the invariants are derived independently through different OCPs, only ω_τ and v_t are, as expected, the components that differ.

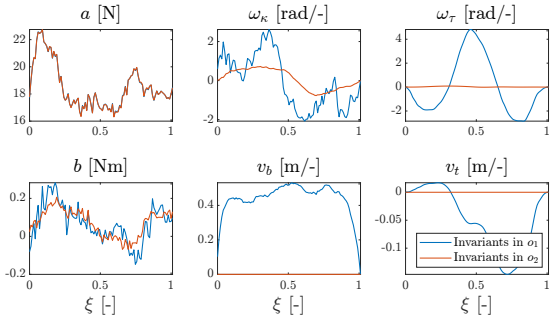


Figure G.15: Screw invariants of wrench derived from wrench data expressed in two viewpoints. As expected, only the object invariant a and b appear consistent across viewpoints.

The two moving frames of motion and the two of wrench, each derived independently in their respective viewpoints, are therefore unsurprisingly unaligned. Consequently, their ISAs are not directly related, resulting in significant differences between the moving frames, as illustrated in Figures G.16 and G.17. Calculation of the shortest distance between the two frames causes the discontinuities in Figure G.16.

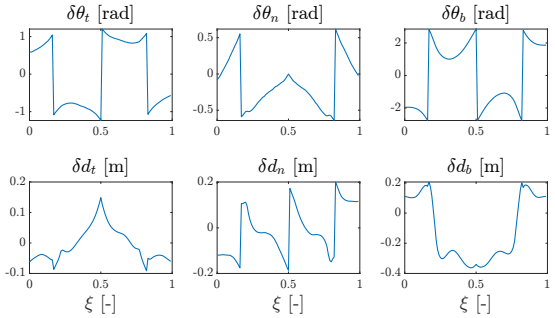


Figure G.16: The distance between the two ISA frames of motion is calculated by (G.2). Unlike in the synthetic data, these ISA frames here show significant differences.

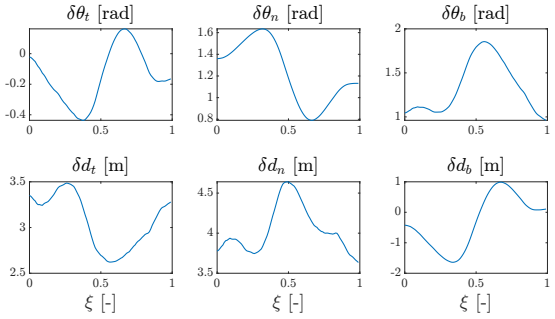


Figure G.17: The distance between the two ISA frames of wrench is calculated by (G.2). Unlike in the synthetic data, these ISA frames exhibit notable differences.

In this case study, a singularity exists in the second viewpoint for wrench invariants. Additionally, the imposed desired tolerances in the OCP influence the ISA's behavior of motion and wrench. Without constraints that impose a direct relation between the moving frames across the two viewpoints, the expected relations, particularly at the pose level for both the frames and the ISA, are not clearly observable.

Appendix H

Discretized Analytical Formulas for Calculation of Screw Trajectory from Screw Invariants

Aim of this appendix is to find a screw trajectory from screw invariants. So, the input is object invariants $\widetilde{\boldsymbol{\mathfrak{z}}}_k$ and moving frame invariants $\dot{\boldsymbol{\imath}}_k$ with $k = 1 \dots N$ and progress step $\Delta\xi$. The invariants are as follows according to Table 2.3.

$$\widetilde{\boldsymbol{\mathfrak{z}}}_k = (a_k \;\; 0 \;\; 0 \;\; b_k \;\; 0 \;\; 0)^{\mathrm{T}} \tag{H.1}$$

$$\dot{\boldsymbol{\imath}}_k = (\omega_{\tau\,k} \;\; 0 \;\; \omega_{\kappa\,k} \;\; v_{t\,k} \;\; 0 \;\; v_{b\,k})^{\mathrm{T}} \tag{H.2}$$

Based on to (2.42), ISA frames can be calculated as follows,

$$\widetilde{\boldsymbol{T}}_{k+1} = \widetilde{\boldsymbol{T}}_k \exp\left([\dot{\boldsymbol{\imath}}_k]_\times \, \Delta\xi\right), \tag{2.42}$$

where $\dot{\boldsymbol{\imath}}_k$ represents the moving frame invariants at sample k as a skew symmetric matrix according to (2.41) and (2.4).

The object invariant can be transformed to world reference frame using (2.38) as follows. This shows the reconstructed screw trajectory.

$$\boldsymbol{\mathfrak{z}}_k = \boldsymbol{S}(\widetilde{\boldsymbol{T}}_k) \; \widetilde{\boldsymbol{\mathfrak{z}}}_k \tag{H.3}$$

If $\boldsymbol{\mathfrak{z}} = \boldsymbol{t}$, we can also find object pose trajectory by the following equation using (2.6):

$$\boldsymbol{T}_{k+1} = \exp\left([\boldsymbol{\mathfrak{z}}_k]_\times \Delta\xi\right) \boldsymbol{T}_k \tag{H.4}$$

Clearly, if $\boldsymbol{\mathfrak{z}} = \boldsymbol{w}$ or we are only interested in object twist, $\boldsymbol{t}$, we do not need to perform the last step.

Appendix I

Discretized Analytical Formulas for Calculation of Screw Invariants from Screw Trajectory*

Aim of this appendix is to find screw invariants from a screw trajectory. So, the input is a screw trajectory $\boldsymbol{\jmath}$ with $k = 1 \dots N$ and progress step $\Delta\xi$. This is shown at sample k:

$$\boldsymbol{\jmath}_k = (\boldsymbol{a}_k \quad \boldsymbol{b}_k)^T. \tag{I.1}$$

Discretized version of unit vectors of a ISA frame are as follows using (2.14):

$$\boldsymbol{e}_{t\,k} = \frac{\boldsymbol{a}_k}{\|\boldsymbol{a}_k\|}, \quad \boldsymbol{e}_{n\,k} = \frac{(\boldsymbol{a}_k \times \boldsymbol{a}_{k+1}) \times \boldsymbol{a}_k}{\|(\boldsymbol{a}_k \times \boldsymbol{a}_{k+1}) \times \boldsymbol{a}_k\|}, \quad \boldsymbol{e}_{b\,k} = \boldsymbol{e}_{t\,k} \times \boldsymbol{e}_{n\,k}. \tag{I.2}$$

While $\boldsymbol{e}_{b\,k}$ can be written as follows according to (E.15).

$$\boldsymbol{e}_{b\,k} = \frac{\boldsymbol{a}_k \times \boldsymbol{a}_{k+1}}{\|\boldsymbol{a}_k \times \boldsymbol{a}_{k+1}\|} \tag{I.3}$$

The point $\widetilde{\boldsymbol{p}}_\perp$ on the ISA closest to the origin of the reference frame is determined as shown in (2.33) for two consecutive steps:

$$\widetilde{\boldsymbol{p}}_{\perp\,k} = \frac{\boldsymbol{a}_k \times \boldsymbol{b}_k}{\|\boldsymbol{a}_k\|^2} \tag{I.4}$$

*The material in this appendix is based on an internal report by Maxim Vochten.

$$\widetilde{\boldsymbol{p}}_{\perp k+1} = \frac{\boldsymbol{a}_{k+1} \times \boldsymbol{b}_{k+1}}{\|\boldsymbol{a}_{k+1}\|^2} \tag{I.5}$$

According to [4], given two axes defined by points $\widetilde{\boldsymbol{p}}_{\perp k}$ and $\widetilde{\boldsymbol{p}}_{\perp k+1}$ with directions $\boldsymbol{\omega}_k$ and $\boldsymbol{\omega}_{k+1}$, respectively, the points on each axis closest to the other, corresponding to the origins of the ISA frames ($\widetilde{\boldsymbol{p}}_k$ and $\widetilde{\boldsymbol{p}}_{k+1}$), can be determined. In other words, we can find the origin of the moving frame that intersects the common normal, as illustrated in Figure I.1.

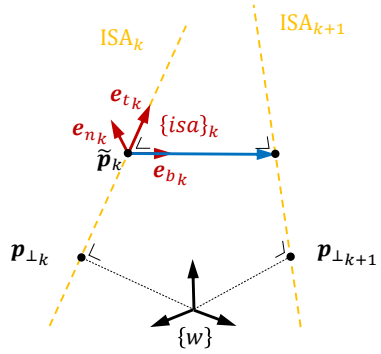


Figure I.1: Two consecutive instantaneous screw axes (ISAs) and the common normal between them, used to determine the ISA frame origins.

Thus, ISA frames for all samples can be constructed as follows using (I.2) and (2.36):

$$\widetilde{\boldsymbol{T}}_k = \begin{pmatrix} \boldsymbol{e}_{t\,k} & \boldsymbol{e}_{n\,k} & \boldsymbol{e}_{b\,k} & \widetilde{\boldsymbol{p}}_k \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{I.6}$$

The object invariants are derived as follows as explained in (2.37).

$$a_k = \boldsymbol{a}_k \cdot \boldsymbol{e}_{t\,k} \tag{I.7}$$

$$b_k = \boldsymbol{b}_k \cdot \boldsymbol{e}_{t\,k} \tag{I.8}$$

Since the trajectory of the ISA frame is known, we can find the moving frame invariants as follows using (2.42):

$$\begin{pmatrix} 0 & -\delta\theta_b & \delta\theta_n & \delta d_t \\ \delta\theta_b & 0 & -\delta\theta_t & \delta d_n \\ -\delta\theta_n & \delta\theta_t & 0 & \delta d_b \\ 0 & 0 & 0 & 0 \end{pmatrix} = \log(\widetilde{\boldsymbol{T}}_k^{\,-1}\, \widetilde{\boldsymbol{T}}_{k+1}) \tag{I.9}$$

$$\omega_{\kappa\,k} \approx \frac{\delta\theta_b}{\Delta\xi} \tag{I.10}$$

$$\omega_{\tau\,k} \approx \frac{\delta\theta_t}{\Delta\xi} \tag{I.11}$$

$$v_{t\,k} \approx \frac{\delta d_t}{\Delta\xi} \tag{I.12}$$

$$v_{b\,k} \approx \frac{\delta d_b}{\Delta\xi} \tag{I.13}$$

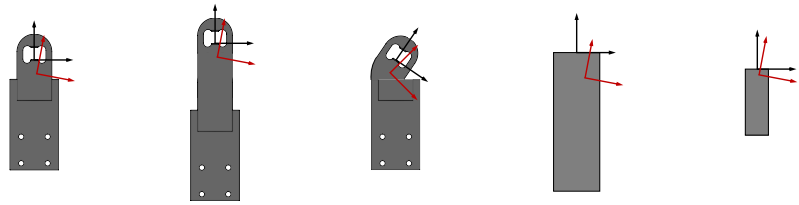
Appendix J

Generalization Across New Objects

In this appendix, we illustrate how identifying an object's key feature can assist in determining an appropriate frame attached to the object of interest. While the selection may seem trivial in some cases, a well-chosen frame significantly simplifies the generalization process across new objects.

We begin by reviewing two examples. In the Bottle-Opening task, the lower tip of the bottle opener serves as a crucial feature, while for the bottle, the tip of its cap is the key feature. Similarly, in the Peg-on-Hole Alignment task, the tip of the peg and the top center of the hole are identified as important features. These features can be leveraged to define object frames, which in turn enable the definition of the task frame relative to the object frame.

Figure J.1 illustrates three simplified cases. Figures J.1a and J.1d show the tools used during data collection, where the object frame is defined based on a key feature of the tool, and the task frame is defined relative to the object frame. Figures J.1b, J.1c, and J.1e illustrate how the object frame adapts to geometric changes by translating, rotating, or remaining unchanged, respectively. This adaptive behavior ensures that the frame stays aligned with the object's key feature. As a result, the task frame can adjust relative to the object frame across varying scenarios: unchanged, rotated, or scaled in position.



(a) (b) (c) (d) (e)

Figure J.1: Task frames derived from data and their adapted versions for new objects are illustrated, taking into account the objects' specific features. The task frame and object frame are shown in *black* and *black*, respectively. For the Bottle-Opening task, an object frame is defined at the opener's lower tip. (a) The task frame derived from data is expressed relative to the object frame. (b) Configuration of frames with a new longer bottle opener. The object and task frames are only translated. (c) Configuration of frames with a new rotated bottle opener. The object and task frames are only rotated. For the Peg-on-Hole-Alignment task, an object frame is defined at the peg's tip. (d) The task frame derived from data is expressed relative to the object frame. (e) Configuration of frames with a shorter and thinner peg. The object frame remains positioned at the peg's tip, while the task frame position is scaled according to the ratio of the peg diameters as described in Section 3.6.2.

Bibliography

[1] Ageing europe - statistics on population developments. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Ageing_Europe_-_statistics_on_population_developments. Accessed: 2025-05-28.

[2] Ageing europe - statistics on working and moving into retirement. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Ageing_Europe_-_statistics_on_working_and_moving_into_retirement. Accessed: 2025-05-28.

[3] Bullet, real-time physics simulation. https://pybullet.org/wordpress/. Accessed: 2023-12-19.

[4] Closest point between two rays. https://palitri.com/vault/stuff/maths/Rays%20closest%20point.pdf. Accessed: 2025-01-24.

[5] Mujoco, advanced physics simulation. https://mujoco.org/. Accessed: 2023-12-19.

[6] Research project echord++, 3dssc. https://echord.eu/3dssc/index.php.html. Accessed: 2025-05-28.

[7] Invariant trajectory descriptors for motion and force, 2023.

[8] Abu-Dakka, F. J., Rozo, L., and Caldwell, D. G. Force-based variable impedance learning for robotic manipulation. *Robotics and Autonomous Systems 109* (2018), 156–167.

[9] Aertbeliën, E., and De Schutter, J. etasl/etc: A constraint-based task specification language and robot controller using expression graphs. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2014), IEEE, pp. 1540–1546.

[10] Ancillao, A., Vochten, M., Verduyn, A., De Schutter, J., and Aertbeliën, E. An optimal method for calculating an average screw axis for a joint, with improved sensitivity to noise and providing an analysis of the dispersion of the instantaneous axes. *PLOS ONE 17*, 10 (2022), 1–22.

[11] Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation 11*, 1 (2019), 1–36.

[12] Arn, R. T., Narayana, P., Emerson, T., Draper, B. A., Kirby, M., and Peterson, C. Motion segmentation via generalized curvatures. *IEEE transactions on pattern analysis and machine intelligence 41*, 12 (2018), 2919–2932.

[13] Atkeson, C. G., An, C. H., and Hollerbach, J. M. Estimation of inertial parameters of manipulator loads and links. *The International Journal of Robotics Research 5*, 3 (1986), 101–119.

[14] Bahety, A., Mandikal, P., Abbatematteo, B., and Martín-Martín, R. Screwmimic: Bimanual imitation from human videos with screw space projection. In *Robotics: Science and Systems (RSS), 2024* (2024).

[15] Ball, R. S. *A Treatise on the Theory of Screws*. Cambridge university press, 1900 (republished 1998).

[16] Barrau, A., and Bonnabel, S. The invariant extended kalman filter as a stable observer. *IEEE Transactions on Automatic Control 62*, 4 (2016), 1797–1812.

[17] Belta, C., and Kumar, V. Euclidean metrics for motion generation on se(3). *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 216*, 1 (2002), 47–60.

[18] Bennequin, D., Fuchs, R., Berthoz, A., and Flash, T. Movement timing and invariance arise from several geometries. *PLOS Computational Biology 5*, 7 (07 2009), 1–27.

[19] Billard, A. G., Calinon, S., and Dillmann, R. *Springer Handbook of Robotics*. Springer, Cham, Switzerland, 2016, ch. Learning from humans, pp. 1995–2014.

[20] Binglong, W., Daokui, Q., Xu, F., Song, J., and Hu, J. A multi-parameter overall identification method used for industrial robot force signal processing [c]. In *Proceeding of the 11th World Congress on Intelligent Control and Automation* (2014), pp. 2437–2441.

[21] Bishop, R. L. There is more than one way to frame a curve. *American Mathematical Monthly* (1975), 246–251.

[22] Bonnabel, S. Left-invariant extended kalman filter and attitude estimation. In *2007 46th IEEE Conference on Decision and Control* (2007), pp. 1027–1032.

[23] Bruyninckx, H., and De Schutter, J. Specification of force-controlled actions in the" task frame formalism"-a synthesis. *IEEE transactions on robotics and automation 12*, 4 (1996), 581–589.

[24] Bruyninckx, H., Demey, S., Dutre, S., and De Schutter, J. Kinematic models for model-based compliant motion in the presence of uncertainty. *The International journal of robotics research 14*, 5 (1995), 465–482.

[25] Bruyninckx, H., Dutre, S., and De Schutter, J. Peg-on-hole: a model based solution to peg and hole alignment. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation* (1995), vol. 2, IEEE, pp. 1919–1924.

[26] Bruyninckx, H., and Khatib, O. Gauss' principle and the dynamics of redundant and constrained manipulators. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)* (2000), vol. 3, pp. 2563–2568 vol.3.

[27] Calabi, E., Olver, P. J., Shakiban, C., Tannenbaum, A., and Haker, S. Differential and numerically invariant signature curves applied to object recognition. *International Journal of Computer Vision 26*, 2 (Feb 1998), 107–135.

[28] Calinon, S. *Learning from Demonstration (Programming by Demonstration)*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2018, pp. 1–8.

[29] Calinon, S., Guenter, F., and Billard, A. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 37*, 2 (2007), 286–298.

[30] Chavez, F. J. A., Traversaro, S., Pucci, D., and Nori, F. Model based in situ calibration of six axis force torque sensors. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)* (2016), IEEE, pp. 422–427.

[31] Conkey, A., and Hermans, T. Learning task constraints from demonstration for hybrid force/position control. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)* (2019), IEEE, pp. 162–169.

[32] Coope, I. D. Circle fitting by linear and nonlinear least squares. *Journal of Optimization theory and applications 76* (1993), 381–388.

[33] De Laet, T., Bellens, S., Smits, R., Aertbelien, E., Bruyninckx, H., and De Schutter, J. Geometric relations between rigid bodies (part 1): Semantics for standardization. *IEEE Robotics Automation Magazine 20*, 1 (March 2013), 84–93.

[34] De Schutter, J. Invariant description of rigid body motion trajectories. *Journal of Mechanisms and Robotics 2*, 1 (2010), 011004/1–9.

[35] De Schutter, J., De Laet, T., Rutgeerts, J., Decré, W., Smits, R., Aertbeliën, E., Claes, K., and Bruyninckx, H. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *The International Journal of Robotics Research 26*, 5 (2007), 433–455.

[36] De Schutter, J., Torfs, D., Bruyninckx, H., and Dutré, S. Invariant hybrid force/position control of a velocity controlled robot with compliant end effector using modal decoupling. *The International Journal of Robotics Research 16*, 3 (1997), 340–356.

[37] De Schutter, J., and Van Brussel, H. Compliant robot motion i. a formalism for specifying compliant motion tasks. *The International Journal of Robotics Research 7*, 4 (1988), 3–17.

[38] De Schutter, J., and Van Brussel, H. Compliant robot motion ii. a control approach based on external control loops. *The International Journal of Robotics Research 7*, 4 (1988), 18–33.

[39] Deniša, M., Gams, A., Ude, A., and Petrič, T. Learning compliant movement primitives through demonstration and statistical generalization. *IEEE/ASME transactions on mechatronics 21*, 5 (2015), 2581–2594.

[40] Duffy, J. The fallacy of modern hybrid control theory that is based on orthogonal complements of twist and wrench spaces. *Journal of Robotic Systems 7*, 2 (1990), 139–144.

[41] El Zaatari, S., Marei, M., Li, W., and Usman, Z. Cobot programming for collaborative industrial tasks: An overview. *Robotics and Autonomous Systems 116* (2019), 162–180.

[42] Featherstone, R. *Rigid body dynamics algorithms*. Springer, 2014.

[43] Fischer, K., Kirstein, F., Jensen, L. C., Krüger, N., Kukliński, K., aus der Wieschen, M. V., and Savarimuthu, T. R. A comparison of types of robot control for programming by demonstration. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (2016), pp. 213–220.

[44] Gasparetto, A., Scalera, L., et al. A brief history of industrial robotics in the 20th century. *Advances in Historical Studies 8* (2019), 24–35.

[45] Gray, A., Abbena, E., Salamon, S., et al. *Modern differential geometry of curves and surfaces with Mathematica*, 3rd ed. CRC press, 2006.

[46] Guggenheimer, H. *Differential Geometry*. McGraw-Hill series in higher mathematics. Dover Publications, 1977.

[47] Guo, Y., Li, Y., and Shao, Z. Rrv: A spatiotemporal descriptor for rigid body motion recognition. *IEEE transactions on cybernetics 48*, 5 (2017), 1513–1525.

[48] Guo, Y., Li, Y., and Shao, Z. Dsrf: A flexible trajectory descriptor for articulated human action recognition. *Pattern Recognition 76* (2018), 137–148.

[49] Hazarika, S. M., and Dixit, U. S. *Robotics: History, Trends, and Future Directions*. Springer International Publishing, Cham, 2018, pp. 213–239.

[50] Huang, Q., Zhang, X., Pan, H., Cheng, Y., and Lu, K. Offsets and gravity parameters estimation for non-contact force compensation with wrist-mounted force/torque sensor. *IEEE Sensors Journal* (2024).

[51] Huang, Y., Silvério, J., Rozo, L., and Caldwell, D. G. Generalized task-parameterized skill learning. In *2018 IEEE international conference on robotics and automation (ICRA)* (2018), IEEE, pp. 5667–5474.

[52] Huynh, D. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision 35* (10 2009), 155–164.

[53] Ijspeert, A. J., Nakanishi, J., Hoffmann, H., Pastor, P., and Schaal, S. Dynamical movement primitives: Learning attractor models for motor behaviors. *Neural Computation 25* (2013), 328–373.

[54] Kavan, L., Collins, S., O'Sullivan, C., and Zara, J. Dual quaternions for rigid transformation blending. *Trinity College Dublin 5* (2006).

[55] Kim, Y., Sloth, C., and Kramberger, A. A framework for transferring surface finishing skills to new surface geometries. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2022), IEEE, pp. 7650–7655.

[56] Kim, Y., Sloth, C., and Kramberger, A. Skill transfer for surface finishing tasks based on estimation of key parameters. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)* (2022), pp. 2148–2153.

[57] Kober, J., Gienger, M., and Steil, J. J. Learning movement primitives for force interaction tasks. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (2015), IEEE, pp. 3192–3199.

[58] Kroemer, O., Niekum, S., and Konidaris, G. A review of robot learning for manipulation: Challenges, representations, and algorithms. *Journal of Machine Learning Research 22*, 30 (2021), 1–82.

[59] Kubus, D., Kroger, T., and Wahl, F. M. Improving force control performance by computational elimination of non-contact forces/torques. In *2008 IEEE international conference on robotics and automation* (2008), IEEE, pp. 2617–2622.

[60] Kwon, J., Choi, K., and Park, F. C. Kinodynamic model identification: A unified geometric approach. *IEEE Transactions on Robotics 37*, 4 (2021), 1100–1114.

[61] Laha, R., Sun, R., Wu, W., Mahalingam, D., Chakraborty, N., Figueredo, L. F., and Haddadin, S. Coordinate invariant user-guided constrained path planning with reactive rapidly expanding plane-oriented escaping trees. In *2022 International Conference on Robotics and Automation (ICRA)* (2022), pp. 977–984.

[62] Lee, D., Soloperto, R., and Saveriano, M. Bidirectional invariant representation of rigid body motions and its application to gesture recognition and reproduction. *Autonomous Robots 42*, 1 (Jan. 2018), 125–145.

[63] Li, X., Baurn, M., and Brock, O. Augmentation enables one-shot generalization in learning from demonstration for contact-rich manipulation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2023), IEEE, pp. 3656–3663.

[64] Li, X., and Brock, O. Learning from demonstration based on environmental constraints. *IEEE Robotics and Automation Letters 7*, 4 (2022), 10938–10945.

[65] Lin, Z., Xin, W., Yang, J., QingPei, Z., and ZongJie, L. Dynamic trajectory-tracking control method of robotic transcranial magnetic stimulation with end-effector gravity compensation based on force sensors. *Industrial Robot: An International Journal 45*, 6 (2018), 722–731.

[66] Lipkin, H. *GEOMETRY AND MAPPINGS OF SCREWS WITH APPLICATIONS TO THE HYBRID CONTROL OF ROBOTIC MANIPULATORS*. PhD thesis, University of Florida, 1985.

[67] Lipkin, H., and Duffy, J. Hybrid twist and wrench control for a robotic manipulator. *Journal of Mechanisms, Transmissions, and Automation in Design 110*, 2 (06 1988), 138–144.

[68] Lynch, K. M., and Park, F. C. *Modern robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.

[69] Mahalingam, D., and Chakraborty, N. Human-guided planning for complex manipulation tasks using the screw geometry of motion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)* (2023), pp. 7851–7857.

[70] Manschitz, S., Gienger, M., Kober, J., and Peters, J. Mixture of attractors: A novel movement primitive representation for learning motor skills from demonstrations. *IEEE Robotics and Automation Letters 3*, 2 (2018), 926–933.

[71] Mason, M. T. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics 11*, 6 (1981), 418–432.

[72] Meirovitch, Y., Bennequin, D., and Flash, T. Geometrical invariance and smoothness maximization for task-space movement generation. *IEEE Transactions on Robotics 32*, 4 (2016), 837–853.

[73] Moons, T., Pauwels, E. J., Van Gool, L., and Oosterlinck, A. Foundations of semi-differential invariants. *International Journal of Computer Vision 14*, 1 (1995), 25–47.

[74] Mousavi Mohammadi, A., Vochten, M., Aertbeliën, E., and De Schutter, J. Automatic derivation of an optimal task frame for learning and controlling contact-rich tasks. *arXiv preprint arXiv:2404.01900* (2024).

[75] Mukherjee, D., Gupta, K., Chang, L. H., and Najjaran, H. A survey of robot learning strategies for human-robot collaboration in industrial settings. *Robotics and Computer-Integrated Manufacturing 73* (2022), 102231.

[76] Murray, R. M., Sastry, S. S., and Zexiang, L. *A Mathematical Introduction to Robotic Manipulation*, 1st ed. CRC Press, Inc., Boca Raton, FL, USA, 1994.

[77] Niekum, S., Osentoski, S., Konidaris, G., Chitta, S., Marthi, B., and Barto, A. G. Learning grounded finite-state representations from unstructured demonstrations. *The International Journal of Robotics Research 34*, 2 (2015), 131–157.

[78] Perantoni, G., and Limebeer, D. J. Optimal control of a formula one car on a three-dimensional track—part 1: Track modeling and identification. *Journal of Dynamic Systems, Measurement, and Control 137*, 5 (2015), 051018.

[79] Pérez-D'Arpino, C., and Shah, J. A. C-learn: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (2017), IEEE, pp. 4058–4065.

[80] Peternel, L., Rozo, L., Caldwell, D., and Ajoudani, A. A method for derivation of robot task-frame control authority from repeated sensory observations. *IEEE Robotics and Automation Letters 2*, 2 (2017), 719–726.

[81] Pettinger, A., Alambeigi, F., and Pryor, M. A versatile affordance modeling framework using screw primitives to increase autonomy during manipulation contact tasks. *IEEE Robotics and Automation Letters 7*, 3 (2022), 7224–7231.

[82] Pham, Q. C., and Nakamura, Y. A new trajectory deformation algorithm based on affine transformations. *IEEE Transactions on Robotics 31*, 4 (Aug 2015), 1054–1063.

[83] Piao, Y., Hayakawa, K., and Sato, J. Space-time invariants for recognizing 3D motions from arbitrary viewpoints under perspective projection. *Transactions of the Institute of Electronics, Information and Communication Engineers E89-D*, 7 (2006), 2268–2274.

[84] Qin, F., Xu, D., Zhang, D., and Li, Y. Robotic skill learning for precision assembly with microscopic vision and force feedback. *IEEE/ASME Transactions on Mechatronics 24*, 3 (2019), 1117–1128.

[85] Racca, M., Pajarinen, J., Montebelli, A., and Kyrki, V. Learning in-contact control strategies from demonstration. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2016), IEEE, pp. 688–695.

[86] Raibert, M. H., and Craig, J. J. Hybrid Position/Force Control of Manipulators. *Journal of Dynamic Systems, Measurement, and Control 103*, 2 (06 1981), 126–133.

[87] Rao, C., Yilmaz, A., and Shah, M. View-invariant representation and recognition of actions. *International Journal of Computer Vision 50*, 2 (2002), 203–226.

[88] Ravichandar, H., Polydoros, A. S., Chernova, S., and Billard, A. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems 3* (2020), 297–330.

[89] Sakr, M., Li, Z. J., Van der Loos, H. M., Kulić, D., and Croft, E. A. Quantifying demonstration quality for robot learning and generalization. *IEEE Robotics and Automation Letters 7*, 4 (2022), 9659–9666.

[90] Shao, Z., and Li, Y. Integral invariants for space motion trajectory matching and recognition. *Pattern Recognition 48*, 8 (2015), 2418–2432.

[91] Sorkine-Hornung, O., and Rabinovich, M. Least-squares rigid motion using svd. *Computing 1*, 1 (2017), 1–5.

[92] Spong, M. W., Hutchinson, S., and Vidyasagar, M. *Robot modeling and control*. John Wiley & Sons, 2020.

[93] Suomalainen, M., Abu-Dakka, F. J., and Kyrki, V. Imitation learning-based framework for learning 6-d linear compliant motions. *Autonomous Robots 45*, 3 (2021), 389–405.

[94] Suomalainen, M., Calinon, S., Pignat, E., and Kyrki, V. Improving dual-arm assembly by master-slave compliance. In *2019 International Conference on Robotics and Automation (ICRA)* (2019), IEEE, pp. 8676–8682.

[95] Suomalainen, M., Karayiannidis, Y., and Kyrki, V. A survey of robot manipulation in contact. *Robotics and Autonomous Systems 156* (2022), 104224.

[96] Suomalainen, M., Koivumäki, J., Lampinen, S., Kyrki, V., and Mattila, J. Learning from demonstration for hydraulic manipulators.

In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2018), IEEE, pp. 3579–3586.

[97] Suomalainen, M., and Kyrki, V. Learning compliant assembly motions from demonstration. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2016), IEEE, pp. 871–876.

[98] Suomalainen, M., and Kyrki, V. A geometric approach for learning compliant motions from demonstration. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)* (2017), IEEE, pp. 783–790.

[99] Traversaro, S., Pucci, D., and Nori, F. In situ calibration of six-axis force-torque sensors using accelerometer measurements. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (2015), IEEE, pp. 2111–2116.

[100] Ureche, A. L. P., Umezawa, K., Nakamura, Y., and Billard, A. Task parameterization using continuous constraints extracted from human demonstrations. *IEEE Transactions on Robotics 31*, 6 (2015), 1458–1471.

[101] Verduyn, A., Vochten, M., and De Schutter, J. Enhancing motion trajectory segmentation of rigid bodies using a novel screw-based trajectory-shape representation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)* (2024), IEEE, pp. 7179–7185.

[102] Vergara Perico, C. A., De Schutter, J., and Aertbeliën, E. Combining imitation learning with constraint-based task specification and control. *IEEE Robotics and Automation Letters 4*, 2 (2019), 1892–1899.

[103] Vergara Perico, C. A., De Schutter, J., and Aertbeliën, E. Learning robust manipulation tasks involving contact using trajectory parameterized probabilistic principal component analysis. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020), IEEE, pp. 8336–8343.

[104] Vochten, M., De Laet, T., and De Schutter, J. Comparison of rigid body motion trajectory descriptors for motion representation and recognition. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (2015), pp. 3010–3017.

[105] Vochten, M., De Laet, T., and De Schutter, J. Comparison of rigid body motion trajectory descriptors for motion representation and recognition. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (May 2015), pp. 3010–3017.

[106] Vochten, M., De Laet, T., and De Schutter, J. Robust optimization-based calculation of invariant trajectory representations for point and rigid-body motion. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2018), pp. 5598–5605.

[107] Vochten, M., De Laet, T., and De Schutter, J. Generalizing demonstrated motion trajectories using coordinate-free shape descriptors. *Robotics and Autonomous Systems 122* (2019), 103291.

[108] Vochten, M., Mousavi Mohammadi, A., Verduyn, A., De Laet, T., Aertbeliën, E., and De Schutter, J. Invariant descriptors of motion and force trajectories for interpreting object manipulation tasks in contact. *IEEE Transactions on Robotics 39*, 6 (2023), 4892–4912.

[109] Vougioukas, S. Bias estimation and gravity compensation for force-torque sensors. In *Proceedings of International Conference on Mathematical Methods and Computational Techniques in Electrical Engineering* (2001), Citeseer, pp. 82–85.

[110] Wächter, A., and Biegler, L. T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming 106*, 1 (2006), 25–57.

[111] Wang, N., Zhou, J., and Zhang, X. Research on the estimation of sensor bias and parameters of load based on force-feedback. In *Intelligent Robotics and Applications: 11th International Conference, ICIRA 2018, Newcastle, NSW, Australia, August 9–11, 2018, Proceedings, Part I 11* (2018), Springer, pp. 404–413.

[112] Weiss, I. Geometric invariants and object recognition. *International Journal of Computer Vision 10*, 3 (1993), 207–231.

[113] Werling, M., Gröll, L., and Bretthauer, G. Invariant trajectory tracking with a full-size autonomous road vehicle. *IEEE Transactions on Robotics 26*, 4 (2010), 758–765.

[114] Wu, S., and Li, Y. Motion trajectory reproduction from generalized signature description. *Pattern Recognition 43*, 1 (2010), 204–221.

[115] Wu, S., and Li, Y. F. On signature invariants for effective motion trajectory recognition. *The International Journal of Robotics Research 27*, 8 (2008), 895–917.

[116] Wu, S., and Li, Y. F. Flexible signature descriptions for adaptive motion trajectory representation, perception and recognition. *Pattern Recognition 42*, 1 (2009), 194–214.

[117] Yang, X., Li, F., Gao, R., Song, R., and Li, Y. Force perception of industrial robot based on multi-parameter coupled model. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (2019), IEEE, pp. 1676–1681.

[118] Yu, Y., Shi, R., and Lou, Y. Bias estimation and gravity compensation for wrist-mounted force/torque sensor. *IEEE Sensors Journal 22*, 18 (2021), 17625–17634.

[119] Zhao, J., Giammarino, A., Lamon, E., Gandarias, J. M., De Momi, E., and Ajoudani, A. A hybrid learning and optimization framework to achieve physically interactive tasks with mobile manipulators. *IEEE Robotics and Automation Letters 7*, 3 (2022), 8036–8043.

List of Publications

Journal papers (peer reviewed)

- Vochten, M., Mousavi Mohammadi, A., Verduyn, A., De Laet, T., Aertbeliën, E., and De Schutter, J. Invariant descriptors of motion and force trajectories for interpreting object manipulation tasks in contact, *IEEE Transactions on Robotics 39*, 6 (2023), 4892-4912.
- Mousavi Mohammadi, A., Vochten, M., Aertbeliën, E., and De Schutter, J. Automatic derivation of an optimal task frame for learning and controlling contact-rich tasks, *arXiv preprint*, (2024).

Conference papers (peer reviewed)

- Mousavi Mohammadi, A., Vochten, M., De Schutter, J., and Aertbeliën, E. Estimating force/torque sensor offsets and gravity parameters using only wrench measurements to facilitate human demonstration of robot manipulation tasks in contact, Accepted for publication in *IEEE International Conference on Automation Science and Engineering (CASE)*, (2025).

Abstracts/Presentations/Posters

- Mousavi Mohammadi, A., Vochten, M., Vanroye, L., Verduyn, A., Ancillao, A., Decré, W., Aertbeliën, E., De Schutter, J. Extending invariant representations of rigid-body motion trajectories towards contact tasks, *Presented at the 1st Flanders Make Scientific Conference*, (2021).

- Vochten, M., Verduyn, A., Mousavi Mohammadi, A., Wellemans, J., Mahy, W., Decré, W. Programming robotic paint spraying tasks by improving human demonstrations with model-based off-line simulations, *Presented at IROS 2022 Workshop on Robotic Systems Integration for Supply Chain Workflows: Design, Deploy, Execute*, Kyoto, Japan, (2022).

- Aertbeliën, E., Mousavi Mohammadi, A. Experiments with sensor based reactive robot control using the constraint based methodology (showcasing eTaSL implementation), *Presented at the 2nd Flanders Make Scientific Conference*, (2022).

- Mousavi Mohammadi, A., De Schutter, J., Aertbeliën, E. Derivation of task frame specifications from passive observations of motion-in-contact human demonstrations, *Presented at the 3rd Flanders Make Scientific Conference*, (2023).

- Mousavi Mohammadi, A., Vochten, M., Aertbeliën, E., De Schutter, J. Automatic extraction of a task frame from human demonstrations for controlling robotic contact tasks, *Presented at the 43rd Benelux Meeting on Systems and Control*, Blankenberge, Belgium, (2024).



FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF MECHANICAL ENGINEERING
ROBOTICS RESEARCH GROUP
Celestijnenlaan 300 box 2402
B-3001 Leuven
seyyedali.mousavimohammadi@kuleuven.be
https://www.mech.kuleuven.be/en/pma/research/robotics