UNIVERSITÉ DE STRASBOURG

ÉCOLE DOCTORALE MSII

Laboratoire ICube, Strasbourg, France

THÈSE présentée par :

Thibault POIGNONEC

Soutenue le : **3 mai 2023**

pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Discipline/ Spécialité : Robotique

Apprentissage automatique pour l'assistance aux gestes chirurgicaux robotisés

"Online models learning for adaptive assistance to teleoperated minimally invasive surgery"

THÈSE dirigée par :

M. BAYLE Bernard	Prof., Université de Strasbourg
M. NAGEOTTE Florent	MdC, Université de Strasbourg

RAPPORTEURS :

M. LELEVE Arnaud	Prof., INSA Lyon
M. ROBUFFO GIORDANO Paolo	DR CNRS, INRIA Rennes

AUTRES MEMBRES DU JURY :

M. CATTIN Philippe	Prof., Université de Bâle (Suisse)
Mme ESSERT Caroline	Prof., Université de Strasbourg
M. ZEMITI Nabil	MdC, Université de Montpellier

Remerciements

Avec cette thèse se terminent plusieurs années passionnantes, parfois éprouvantes, mais aussi transformatrices tant professionnellement que personnellement.

Tout d'abord, je tiens à exprimer ma gratitude à Prof. Arnaud Lelevé et Dr Paolo Robuffo Giordano pour leurs précieux retours, critiques et suggestions qui ont permis d'améliorer la qualité de ce travail. Également, avec Prof. Caroline Essert et Prof. Philippe Cattin, ils m'ont fait l'honneur d'accepter de faire partie du jury de ma thèse et je leur en suis sincèrement reconnaissant.

Je remercie également de tout cœur toutes les personnes qui m'ont aidé au jour le jour pendant mon doctorat, à commencer par mes encadrants Bernard Bayle, Florent Nageotte et Nabil Zemiti. En plus de leur grande compétence et de la qualité d'encadrement, tous trois mixent très bien les échanges professionnels et amicaux, rendant cette période de collaboration très plaisante. Je voudrais exprimer une gratitude toute particulière envers Florent Nageotte qui m'a accompagné dans mes recherches depuis mon premier jour au laboratoire ICube. Ses conseils, ainsi que ses encouragements et sa bienveillance, ont grandement contribué à la réussite de cette thèse.

Un grand merci à Philippe Zanne pour avoir partagé son expertise sur les endoscopes flexibles et grandement contribué à la réalisation de mes expériences sur des endoscopes, notamment la plateforme STRAS. Hassan Omran, Andrea Cherubini et Paolo Robuffo Giordano ont également contribué à la maturation scientifique de ce projet par des échanges lors des comités de suivi de thèse auxquels ils ont participé. De même, je remercie chaleureusement Benoit Rosa pour ses retours sur les travaux menés sur l'estimation du jeu mécanique et ses retours précieux sur l'une des publications. Je remercie également Robin Passama et Benjamin Navarro pour avoir partagé leur expertise et avoir répondu à mes (très nombreuses) questions concernant les logiciels développés au sein du laboratoire LIRMM.

De nombreuses personnes, que ce soit autour d'un café à la pause, d'un Mlawi à midi ou d'une bière bien méritée en fin de journée, ont égayé et rendu vraiment sympas ces années de doctorat. À la fois au laboratoire ICube, aux côtés de Maciej, Gaëlle, Benoit, Karine, Loïc et bien d'autres, mais aussi au LIRMM aux côtés de Lucas, Lenaic, Joao, Felix ou Adrien, j'ai passé de très bons moments et je les en remercie. Petite dédicace particulière à John Berre et Benjamin Calmé avec qui j'ai commencé cette aventure quand nous étions tous les trois stagiaires et partagions un coin de table à l'IRCAD.

De nombreuses autres personnes ont participé de près ou de loin à la réussite de cette thèse, à commencer par ma famille. Bien que toutes ne peuvent être mentionnées sur une page, elles ont ma sincère gratitude.

Enfin, je souhaite remercier Claire pour sa patience infinie durant ces dernières années où ma thèse a occupé nombre de mes soirées. Par-dessus tout, je suis infiniment reconnaissant qu'elle soit présente dans ma vie.

Abstract – Résumé

Abstract

This thesis explores assistive robotic behaviors for the realization of robot-assisted minimally invasive surgery (MIS), with a focus on laparoscopy and flexible endoscopy. In the first part of this document, we present novel approaches to perform the online backlash model identification of cable-actuated flexible endoscopes. Backlash in cable transmission degrades open-loop positioning accuracy and increases the cognitive load of the practitioner. However, its compensation requires an accurate model identification, which should ideally be done *in-situ*, i.e., just before or even during the surgical procedure. We propose several methods that can be applied to different relevant robot architectures and scenarios. The algorithms are evaluated in simulation and implemented on robotized endoscopic platforms for experimental validation. In the second part of this thesis, we investigate the online learning of the task and robot model parameters to continuously improve assistance to the operator using such a learning. We consider the case of haptic guidance during teleoperation, a scenario especially relevant to surgical robotics. In this context, we avoid relying on exteroceptive sensors as the main source of information as they could be limited or intermittently unavailable. Instead, we rely on the presence of the operator to extract the information necessary for learning. The algorithms we propose are evaluated in different simulated and real telerobotic scenarios, demonstrating the applicability of the methods to online registration problems.

Keywords:

Medical robotics; Haptic guidance; Registration; Backlash; Online parameters learning.

Résumé

Dans cette thèse, nous développons des stratégies d'assistance à la chirurgie minimalement invasive robotisée et plus précisément à la coelioscopie et à l'endoscopie flexible. Différents défis rendent l'exécution automatique de tâches complexes en MIS : d'une part, l'environnement est non structuré et déformable ; d'autre part, les capteurs extéroceptifs sont limités et leur mesure parfois indisponible. De plus, les outils chirurgicaux utilisés sont souvent flexibles, ce qui rend leur positionnement précis compliqué.

Dans la première partie de ce document, nous présentons de nouvelles approches pour réaliser l'identification en ligne du modèle du jeu mécanique présent dans les transmissions à câble utilisées par les endoscopes flexibles. Le jeu dans les transmissions à câbles dégrade la précision du positionnement en boucle ouverte et augmente la charge cognitive du praticien qui devra le compenser. Cependant, sa compensation par la commande nécessite une identification précise du modèle, qui devrait idéalement être effectuée *in-situ*, c'est-à-dire juste avant ou durant la procédure chirurgicale. Nous proposons plusieurs méthodes qui peuvent être appliquées à différentes architectures de robots et dans différents scénarios pertinents pour des applications médicales. Les algorithmes sont évalués à travers des simulations et évalués expérimentalement sur une plateforme endoscopique robotisée.

Dans une seconde partie, nous étudions l'apprentissage en ligne des paramètres des modèles de la tâche et du robot afin de générer une assistance à l'opérateur qui pourra s'améliorer en cours d'utilisation. Nous considérons le cas du guidage haptique lors de la téléopération à distance d'un robot, un scénario classique en robotique chirurgicale. Dans ce contexte, nous évitons d'être dépendant de capteurs extéroceptifs et exploitons la présence de l'opérateur pour extraire les informations nécessaires à l'apprentissage. Les algorithmes que nous proposons sont évalués dans différents scénarios télérobotiques simulés et réels, démontrant l'applicabilité des méthodes aux problèmes d'apprentissage en ligne pour l'assistance à la téléopération.

Mots-clefs :

Robotique médicale ; Guidage haptique ; Recalage ; Jeu mécanique ; Apprentissage automatique.

N.B. : Un résumé détaillé en français est inclu dans l'annexe F de ce document.

Contents

Abstract – Résumé iii

Contents v

List of abbreviations and frequently used notations x

List of Figures xii

List of Tables xviii

List of complementary materials xix

1 Introduction 1
- 1.1 Robotic assistance for endoscopic surgery . . . 2
  - 1.1.1 Evolution of the surgical practice . . . 2
  - 1.1.2 Robot-assisted minimally invasive surgery . . . 3
  - 1.1.3 Autonomy levels in robot-assisted surgery . . . 5
- 1.2 Assistive robotic behaviors for MIS . . . 7
  - 1.2.1 Assisted robot control . . . 7
  - 1.2.2 Shared control paradigms . . . 9
  - 1.2.3 Automatic sub-task execution . . . 11
- 1.3 Challenges to assistive robotic behaviors for MIS . . . 11
  - 1.3.1 Intra-operative task registrations . . . 12
  - 1.3.2 Inaccurate robot modeling . . . 13
  - 1.3.3 Effects of task and robot model inaccuracies . . . 13
- 1.4 Content of the thesis . . . 15
  - 1.4.1 Organization of the manuscript . . . 15
  - 1.4.2 Contributions and scientific communications . . . 16

I In-situ backlash estimation for robotized flexible endoscopy 18

2 A Bayesian approach for online endoscopic robots model identification 19
- 2.1 Related work . . . 20
  - 2.1.1 Structure of cable-driven flexible endoscopes . . . 20
  - 2.1.2 Backlash modeling and estimation . . . 21
  - 2.1.3 *In-situ* backlash estimation . . . 22
- 2.2 Proposed approach . . . 23
  - 2.2.1 Robot kinematic modeling . . . 24
  - 2.2.2 State and parameters estimation . . . 25

2.3 One DOF illustration . . . 28
2.3.1 One-dimensional simulation . . . 28
2.3.2 Comparison with classical EKF . . . 30
2.4 Simulation results . . . 31
2.4.1 Simulation setup . . . 31
2.4.2 Results . . . 33
2.5 Conclusions . . . 33

3 Eye-in-hand endoscopic robot in-situ backlash identification 36
3.1 Preliminary developments for endoscope and backlash modeling . . . 38
3.1.1 Forward kinematics of eye-in-hand endoscopes . . . 38
3.1.2 TSM backlash modeling . . . 38
3.1.3 Backlash model parameterization . . . 39
3.2 Proposed backlash width identification method . . . 40
3.2.1 Problem formulation . . . 40
3.2.2 Comparison with visual SLAM . . . 42
3.2.3 Choice of parameters to be learned . . . 42
3.2.4 Practical filter implementation . . . 43
3.3 Simulation results . . . 44
3.3.1 Simulation setup . . . 44
3.3.2 Backlash width estimation when $\hat{m}_{O,i}$ is correct . . . 45
3.3.3 Robustness to incorrect $\hat{m}_{O,i}$ values . . . 46
3.3.4 Discussion . . . 46
3.4 Experimentation on a robotic endoscopic platform . . . 48
3.4.1 Experimental setup and ground truth estimation . . . 48
3.4.2 Results . . . 50
3.4.3 Discussion . . . 50
3.5 Conclusions . . . 52

4 Backlash width identification from motion detection events 53
4.1 Modeling of the backlash as cascaded non-linearities . . . 55
4.2 Estimation of the pure backlash component . . . 56
4.2.1 Estimation of the backlash at a given motor position . . . 56
4.2.2 Construction of the backlash width function . . . 57
4.2.3 Use for backlash compensation . . . 58
4.3 Remaining function reconstruction . . . 59
4.3.1 General method . . . 59
4.3.2 Proposed implementation for 1 DOF identification . . . 59
4.3.3 Use for improved robot positioning . . . 60
4.4 Experimentation on the STRAS endoscopic platform . . . 60
4.4.1 Experimental setup . . . 60
4.4.2 Method implementation . . . 61
4.4.3 Results . . . 62
4.4.4 Discussion . . . 64
4.5 Conclusions . . . 65

Discussion **67**
Comparison of the *in-situ* backlash estimation approaches 67
Robot modelling and available measurements 67
In-situ VS online 69
Backlash model complexity 69
Practical considerations for the compensation of backlash 70

II Task and robot model correction from human in-the-loop **72**

5 Simultaneous task and robot models learning from user input **73**
5.1 Related work 74
5.2 Problem formulation 77
5.2.1 Considered setup and modeling 77
5.2.2 User desired trajectory 77
5.2.3 Haptic guidance 78
5.2.4 Learning problem 80
5.3 Proposed approach 80
5.3.1 User in-the-loop as an information source 80
5.3.2 General optimization formulation 81
5.3.3 Recursive implementation using an EKF 82
5.3.4 Practical tuning of hyperparameters 83
5.4 Illustrative simulation results 85
5.4.1 Description of the 2D toy scenario 85
5.4.2 Results with simulated user inputs 86
5.4.3 Discussion 88
5.5 Management of variable time parameters 89
5.5.1 Modeling of the time parameters variability 89
5.5.2 Simulation results 90
5.5.3 Alternative time parameterization 91
5.6 Conclusions 92

6 Use-case: Online task and hand-tool registration during haptic guidance **93**
6.1 Description of the considered scenario 94
6.2 Identifiability of the task and robot parameters 97
6.3 Experimental validation of the learning method 98
6.3.1 Learning performance with a human in the loop 98
6.3.2 Influence of the fading factor 103
6.3.3 Robustness to parameters shift 104
6.4 Comparison with non-adaptive guidance 105
6.4.1 Experimental protocol 105
6.4.2 Results 106
6.4.3 Analysis 109
6.4.4 Necessity for joint robot and task models corrections 109
6.5 Conclusions 110

7 Qualitative performance of the adaptive haptic guidance **111**
- 7.1 User-study design . . . 112
  - 7.1.1 Experimental protocol . . . 112
  - 7.1.2 Experimental setup . . . 114
  - 7.1.3 Clutching for mapping re-calibration . . . 114
  - 7.1.4 Collected experimental data . . . 115
- 7.2 Results . . . 117
  - 7.2.1 Objective results across participants . . . 117
  - 7.2.2 Subjective results across participants . . . 118
- 7.3 Analysis . . . 119
  - 7.3.1 Effects of inaccuracies . . . 119
  - 7.3.2 Performance and acceptance of adaptive haptique guidance . . . 120
  - 7.3.3 Perception of guidance compliance . . . 121
- 7.4 Effect of the mapping recalibration . . . 121
- 7.5 Conclusions . . . 123

Discussion **124**
- To guide or not to guide? . . . 124
- Sliding window-based learning – an alternative to EKF . . . 125
  - An adaptive size sliding window learning method for online task registration 125
  - Comparison of EKF with sliding window-based learning . . . 126
- Passivity of adaptive haptic guidance . . . 127

III Final conclusion and perspectives **129**

Conclusion and perspectives **130**
- Conclusion . . . 130
- Perspectives and future work . . . 131
  - Improving the online learning of model parameters . . . 131
  - Simultaneous backlash and task models correction . . . 133
  - Towards more autonomy . . . 133

Appendices **134**

A Detailled list of scientific communications **134**

B Kinematic model of the considered cable-actuated endoscopes **136**
- B.1 Eye-to-hand 3 DOF endoscope (chapters 2 and 4) . . . 136
- B.2 Eye-in-hand 2 DOF endoscope (Chapter 3) . . . 136

C Identifiability analysis of the task and robot parameters (Chapter 6) **138**

D User study (Chapter 7) **142**
- D.1 Complement to Figure 7.2 . . . 142
- D.2 Questionnaire for subjective performance assessment . . . 143

E Alternative implementation of the learning algorithm **144**
E.1 Considered scenario and problem statement 144
E.2 Proposed sliding adaptive-size window-based learning method 145
E.2.1 Learning the task parameters 145
E.2.2 Online window size management 147
E.2.3 Adaptive haptic guidance 147
E.3 Experimentation with an operator in-the-loop 148
E.3.1 Experimental setup 148
E.3.2 Results 149
E.4 Conclusions 152

F Extended abstract (french) **153**
F.1 Introduction 153
F.1.1 La chirurgie minimalement invasive 153
F.1.2 Vers une assistance active à la chirurgie robotisée 154
F.1.3 Objectifs de la thèse 156
F.2 Organisation du document et contributions 157
F.2.1 Identification du jeu mécanique des endoscopes flexibles robotisés 157
F.2.2 Apprentissage de modèles pour la commande partagée 159
F.3 Conclusions 160

Bibliography **162**

List of abbreviations and frequently used notations

Abbreviations

2D	Two-dimensional
3D	Three-dimensional
DEKF	Discontinuous extended Kalman filter (see Chapter 2)
DOF	Degree(s) of freedom
EKF	Extended Kalman filter
HSC	Haptic shared control
MIS	Minimally invasive surgery
NOTES	Natural orifice translumenal endoscopic surgery
OCT	Optical coherence tomography
RMSE	Root mean square error
SLAM	Simultaneous localization and mapping
SSC	State shared control
VF	Virtual fixture(s)
w.r.t.	With respect to

Main symbols

$\mathcal{F}$	Cartesian frame of reference
$\mathcal{K}(\cdot)$	Robot forward kinematic model
$\mathcal{L}(\cdot)$	Real valued cost function
$\mathcal{N}(.)$	Multivariate Gaussian distribution
$\mathcal{X}$	State of a dynamic system
θ	Model parameters

$f(\cdot)$ State transition model (i.e., to model the evolution of $\mathcal{X}$)

$h(\cdot)$ Measurement model (i.e., to model z)

q Motor (Part I) or joint (Part II) position

t_k Discrete time (as it is generally the case in the thesis). Note that when used as an indice, k denotes discrete time (e.g., z_k is a sampled measurement)

x Cartesian pose of the robot

z Measurements

Specific to Part I

$\mathcal{B}(\cdot)$ Backlash model (function with switching conditions)

c Intermediary configuration variable from which the Cartesian robot pose x can be computed. It used to model the unmeasured output a cable transmission

p Cartesian positions of N scene objects w.r.t. the robot or the camera such that $p = \begin{bmatrix} p_0^T & p_1^T \cdots p_N^T \end{bmatrix}^T$ where $p_i \in \mathbb{R}^3$

Specific to Part II

$\mathcal{M}(\cdot)$ Mapping from master to follower operational spaces

θ_g Task model parameters

θ_r Follower robot model parameters

$g(\cdot)$ Model of x_s^d

x_m^g Reference pose for the master side haptic guidance

x_m Cartesian pose of the master robot

x_s Cartesian pose of the follower robot

x_s^d Cartesian pose of the follower robot desired by the operator

x_s^r Reference Cartesian pose tracked by the follower robot controller

List of Figures

1.1 Illustration of the evolution of endoscopic techniques. 2
1.2 Laparoscopic surgery performed manually. 3
1.3 Intuitive Surgical da Vinci robot Si for robot-assisted laparoscopy. . . . 4
1.4 (a) Anubiscope endoscopic system (Dallemagne et al. 2010). (b) NOTES surgery with the Anubiscope (Nageotte et al. 2020). (c) Using a robotic platform (STRAS, Zorn et al. 2018), a single surgeon can control the 10 DOF of the Anubiscope. 5
1.5 Robotized platforms for endoluminal and transluminal surgeries. 5
1.6 Distal end of the Intuitive Surgical SP robot for single-port surgery. . 6
1.7 (a) Illustration of cable actuated bending, adapted from (Bardou et al. 2012). (b) If not compensated, backlash in the transmission results in significant tracking errors in operational space, here with a flexible endoscopic tool of the STRAS platform (Aleluia Porto 2021). 8
1.8 Simplified illustration of the general HSC concept for telemanipulation, see Abbink et al. (2012) for a more detailed representation. 9
1.9 Illustration of *in-situ* task registration errors arising from preoperative to intra-operative registration (a) or errors introduced over time due to tissue tracking inaccuracies (b). 13
1.10 Illustration of an inaccurate automatic task execution resulting from the task and robot modeling errors. 14

2.1 Illustration of cable-driven bending in one plane (i.e., 1 bending DOF). 20
2.2 Illustration of most common analytical hysteresis models for parametric backlash modeling. 21
2.3 The endoscope end-effector pose can be obtained through external tracking (a-b) or *in-situ* pose estimation from endoscopic images (c). Sources (ordered list): Bardou et al. (2012), Aleluia Porto et al. (2019), Reilink et al. (2013b). 22
2.4 Hysteretic behaviors measured on a tendon driven DOF of a flexible endoscopic robot, between proximal actuator and distal configuration. . 23
2.5 Kinematic chain of an n-DOF robot with actuation backlash. 24
2.6 Left: real data illustrating backlash characteristics for three DOF of a robotized endoscope, acquired with the STRAS robotic endoscope (Zorn et al. 2018). Right: backlash model $\mathcal{B}_i(\cdot)$ relating the i^{th} actuator input and output. 25
2.7 Illustration of the effect of the row ordering matrix T. 26

2.8 Left: input actuator position q_k, backlash element output c_k, and noisy measurement z_k used for the simulation. The estimation of c_k computed using the initial model estimation is reported. Right: input-output characteristics of the backlash element from the ground truth model and from the estimated model at $k = 0$. 28
2.9 Top: State variable c_k, its estimation $\hat{c}_{k|k}$ by the FM-DEKF, and estimation errors $\tilde{c}_k = c_k - \hat{c}_{k|k}$. Center: Estimation errors of the state variables associated with the model parameters. Bottom: estimated active branch of the backlash element. 29
2.10 Left: Estimation errors of the state variables associated with the model parameters when using a classical FM-EKF. Right: The variances of the estimates are displayed on a logarithmic scale for the two parameters associated with the branch (L). 30
2.11 Endoscopic robot with 3 DOF, characterized by its axial translation along Z, the bending angle of the bending section of the body, and the rotation w.r.t Z axis. 31
2.12 Input-output characteristics of the backlash elements $\mathcal{B}_i(\cdot)$ from the ground truth models and from the estimated models at $k = 0$. 32
2.13 Top: Ground truth values for $c_{i,k}$. Bottom: FM-DEKF prediction error $\tilde{c}_{i,k} = c_{i,k} - \hat{c}_{i,k}$. 33
2.14 Normalized estimation error for all model parameters computed as the element-wise ratio $\tilde{\mathcal{X}}_k / \tilde{\mathcal{X}}_0$ where $\tilde{\mathcal{X}}_k = \mathcal{X}_k - \hat{\mathcal{X}}_k$. 34

3.1 Considered scenario with an eye-in-hand endoscope. 37
3.2 Left: TSM-actuated endoscopic robot with 2 DOF and an eye-in-hand camera configuration. Right: constant curvature model (Ott et al. 2011). 39
3.3 Kinematic chain of the robot with actuation backlash. 39
3.4 Two equivalent backlash model parameterizations where the variable backlash width (in green) is independent from the overall backlash characteristic (in red). 40
3.5 Intuitions behind identification issues for the absolute orientation and the angular velocity. 43
3.6 Input actuator positions $q_{i,k}$ and backlash element outputs $c_{i,k}$ in mm for each DOF. 45
3.7 Hysteretic cycles resulting from the ground truth (green) and estimation of c_k (red) during the learning process in the case where the estimates of $m_{O,i}$ are correct. The initial model is also represented (blue). 46
3.8 Top: relative backlash model parameter estimation errors in the case where the estimates of $m_{O,i}$ are correct. Bottom: Resulting estimated model $\hat{b}_i(\bar{q}_i)$ and ground truth. 47
3.9 Top: hysteretic cycles resulting from the ground truth and estimation of c_k during the learning process in the case where $m_{O,i}$ and $C_{O,i}$ estimates are incorrect. Bottom: relative backlash model parameter estimation errors in the case where all $m_{O,i}$ and $C_{O,i}$ estimates are incorrect. . . . 48

3.10 Top: RMSE of the landmark position estimation in the case where the estimates of $m_{O,i}$ are correct. Bottom: Case where the $m_{O,i}$ and $C_{O,i}$ estimates are incorrect. 49
3.11 Experimental test bed with a robotized endoscope. 49
3.12 Left: image measurements acquired during a calibration sequence composed of slow, smooth DOF by DOF motions. Right: manual backlash width measurement from calibration sequence and resulting linear regression. 50
3.13 Image measurements collected during the experiments and (posterior) prediction by the FM-DEKF filter. 51
3.14 Top: estimated backlash model parameters and ground truth estimate for the experiment with a real endoscope. Bottom: Resulting estimated model $\hat{b}_i(\bar{q}_i)$ and ground truth. 51

4.1 Proposed modeling of the continuum robot, where transmission nonlinearities are split in two parts. 54
4.2 The backlash (a) is decomposed in a pure backlash component (b) and a remaining function that is possibly non-linear (c). 55
4.3 Pure backlash estimation scheme. 56
4.4 Reconstructed hystereses, by combining a linear f_{coarse} function with pure backlash (top) and by combining estimated f_{remain} with backlash (bottom). 58
4.5 General scheme used for non-linearities estimation. 60
4.6 (a) Backlash width in mm of cable displacement as a function of the mean actuator position. (b) Characteristics between the actuator and configuration variable obtained using configuration variable estimation from images. 62
4.7 Pseudo-periodic trajectory used for testing prediction capability of different models: f_{coarse} (black), $f_{coarse} + B$. (green) and $f_{remain} + B$. (yellow). 63
4.8 Reprojection errors between the measured position of the effector in the endoscopic image and the prediction of effector position provided by f_{coarse} (black), $f_{coarse} + B$. (green) and $f_{remain} + B$. (yellow). 63
4.9 Open loop realization of a reference trajectory (yellow), by relying on f_{coarse}^{-1} (blue) and on the inverse model constructed from $f_{remain} + B$. (see Section 4.3.3) (green). 65
4.10 Left: illustration of the non-linear smooth transition between the deadzone and the branches of the envelope. Right: the model resulting from the identification process is illustrated, both for the case where pure backlash is identified (e.g., Chapter 4) and the case where the functions of the envelope are identified (e.g., Chapter 2). 71

5.1 Block diagram of the position control of the follower robot. 78

5.2 Teleoperation setup where the operator remotely controls the follower robot to execute a desired trajectory (green). An estimation of this desired trajectory (orange) is used to provide a master side haptic guidance, though this estimation is incorrect. The resulting guidance error is compounded by follower robot positioning errors resulting from erroneous kinematic modeling (gray vs. black). 79
5.3 Example of how a robot positioning error can lead to a deformation of the desired master robot trajectory. 80
5.4 Scenario with simulated planar task and robot. 85
5.5 Evolution of the task and robot models at different $t_k \in \{0.5; 5; 10; 20; 50\}s$. See Video 5.1 for full animation. 86
5.6 Parameter estimation over time. 87
5.7 Time parameters estimation and resulting estimation error on $\psi_k = a_k + b_k t_k$ and $\dot{\psi}_k = b_k$. 87
5.8 Evolution of learning metrics. 88
5.9 Simulated pause during the task execution at $t = 50$s. The time parameters (i.e., a and b) estimation becomes inaccurate as the filter does not adapt quickly enough, which introduces significant errors on the other estimated parameters. 89
5.10 Simulated pause during the task execution at $t_k = 50$s. Process noise is injected to cope with variable execution velocity. 91

6.1 Experimental setup. 95
6.2 Overview of the software architecture and main variables. 97
6.3 (a) Illustration of the collected trajectories $x_{s,k}$ and $\hat{x}^d_{s,k}$. (b) The desired trajectory $x^d_{s,k}$ is computed by finding on the ground truth path the points closest to $x_{s,k}$ (for each sample), which then results in errors $||\epsilon_{h,k}||$ (see b) and $||x^d_{s,k} - \hat{x}^d_{s,k}||$ (see c). 99
6.4 Estimated task model evolution over time for the scenario **AG**; the task model is represented as the path underlying $\hat{x}^d_{s,k}$ and all units for X, Y, and Z are in mm. 100
6.5 The norm of the execution (left) and prediction (right) errors under **AG** and **AUTO** are reported for different initial task parameter estimation errors $[\tilde{r}_{z,0}, \tilde{t}_{x,0}, \tilde{t}_{y,0}]$ in [deg., mm, mm]. The initial robot parameter estimation error is $\tilde{L}_0 = 10$ mm. 101
6.6 Top: parameters absolute estimation errors under **AG** and **AUTO** for different initial task parameter estimation errors. Bottom: resulting mean relative errors $\tilde{\theta}_{\mathrm{rel.},k}$. 102
6.7 Average parameter estimation errors as a fraction of its initial value for different values of the fading factor α. The data was collected from 6 optimal executions, i.e., the **AUTO** condition. 103
6.8 Illustrative scenario simulating a parameters shift when the first revolution is completed (around $t_k = 18$ s). 104

6.9 Top: executed, predicted, and desired trajectories are displayed as paths in the drawing plane XY and projected orthogonally (XZ plane) for the three task executions performed by one of the participants. Bottom: associated errors. 106
6.10 Top: the mean relative parameter estimation error $\tilde{\theta}_{\text{rel.},k}$ is averaged over the participants (N=6) for each condition. Bottom: the correspondence between task progress and execution time is displayed. 107
6.11 Mean execution (top) and prediction (bottom) errors for each experimental conditions. 108
6.12 Mean norms of the execution, task prediction, and robot model prediction errors across subjects in mm 108
6.13 Executed, desired, and predicted trajectories for the **AG-TO** scenario. All units for X, Y, and Z are in mm. 110

7.1 Overview of experimental protocol. 113
7.2 Path underlying the trajectories $x_{s,k}$ and $\hat{x}_{s,k}$ expressed in $\mathcal{F}_t$ for one of the participants under the different experimental conditions. 116
7.3 Task duration, execution errors (path deviation), and task and robot model prediction errors ($N = 12$). 117
7.4 Motion smoothness for the different conditions. 118
7.5 NASA TLX scores (lower is better). 119
7.6 Scores obtained for the two additional survey questions. 120
7.7 Collected data about mapping clutching under the different conditions. 122
7.8 Comparison of the execution mean errors over the 2 s time periods before and after the clutch events. 123
7.9 Overview of the sliding window learning approach. 126
7.10 Illustration of the effect of line search local minima on the point matching procedure. 127
7.11 Parameters change detection could be used to modify the properties of the learning process temporarily. 132

C.1 Observability metric $cond(\mathcal{O}(\mathcal{X}))$ computed for different radii of curvature of the task. 141

D.1 Snapshots of the correct and estimated trajectories x_s^d displayed in the XY plane of the drawing frame under condition **AG**. 142

E.1 Illustration of the result of the line search process. 146
E.2 Experimental setup. 149
E.3 Overview of the proposed method and experimental implementation. . 150
E.4 Snapshots of the different positions of interest at different times throughout the experiment. 150
E.5 Top: evolution of the estimated parameters and comparison with the ground truth. Bottom: absolute error on estimated parameters used for guidance $|\theta_{g,i}(t) - \hat{\theta}_{g,i}(t)|$. 151
E.6 Estimated values for $\hat{\psi}(t)$ and $\psi_c(t)$. 152

E.7 Top: criterion γ_k and threshold γ_{max} during the experiment, displayed on a log-scale. Center: zoom of the figure at the top. Bottom: the window shrinks or expands to keep the criterion within a predetermined range. 152

F.1 Illustration de l'évolution des techniques endoscopiques. 153
F.2 Illustration (simplifiée) de la commande partagée haptique. 155
F.3 Illustration des erreurs de recalage *in-situ* d'une tâche robotique. . . . 155
F.4 Illustration de la transmission à câbles utilisée par beaucoup d'endoscopes flexibles et des effets du jeu mécanique. 156
F.5 Illustration des robots endoscopiques considérés. 158
F.6 Illustration du scénario considéré pour les validations expérimentales des chapitres 6 et 7. 160

List of Tables

4.1 Main characteristics of the considered image-based *in-situ* backlash model identification methods for eye-to-hand and eye-in-hand flexible endoscopes. 68

7.1 Additional survey questions for each condition. Translated from French, see original in Appendix D.2. 114

List of complementary materials

Video 5.1 Animation of the simulation presented in Section 5.4.2 (see Figure 5.5).
Url: https://media.hal.science/hal-04107181.

Video 5.2 Animation of the simulation presented in Section 5.5 (see Figure 5.9).
Url: https://media.hal.science/hal-04107375.

Video 5.3 Animation of the simulation presented in Section 5.5.2 (see Figure 5.10).
Url: https://media.hal.science/hal-04107383.

Video 6.1 Online task and hand-tool registration during haptic guidance. Lemniscate drawing task (see Section 6.4), experimental condition **AG**.
Url: https://media.hal.science/hal-04107572.

Video 6.2 Online task and hand-tool registration during haptic guidance. Lemniscate drawing task (see Section 6.4), experimental condition **IG**.
Url: https://media.hal.science/hal-04107664.

Video 6.3 Online task and hand-tool registration during haptic guidance. Lemniscate drawing task (see Section 6.4), experimental condition **NG**.
Url: https://media.hal.science/hal-04107683.

Video 7.1 Realization of the drawing task by a participant (under condition **AG**).
Url: https://media.hal.science/hal-04107816.

Chapter 1

Introduction

Contents

1.1 Robotic assistance for endoscopic surgery 2
  1.1.1 Evolution of the surgical practice 2
  1.1.2 Robot-assisted minimally invasive surgery 3
  1.1.3 Autonomy levels in robot-assisted surgery 5
1.2 Assistive robotic behaviors for MIS 7
  1.2.1 Assisted robot control 7
  1.2.2 Shared control paradigms 9
  1.2.3 Automatic sub-task execution 11
1.3 Challenges to assistive robotic behaviors for MIS 11
  1.3.1 Intra-operative task registrations 12
  1.3.2 Inaccurate robot modeling 13
  1.3.3 Effects of task and robot model inaccuracies 13
1.4 Content of the thesis 15
  1.4.1 Organization of the manuscript 15
  1.4.2 Contributions and scientific communications 16

1.1 Robotic assistance for endoscopic surgery

1.1.1 Evolution of the surgical practice

Until the end of the 1980's, open surgery was the only way to operate a patient: the surgeon would cut the skin and other tissues in order to directly visualize the area to operate. Other medical procedures such as the exploration of natural cavities had already begun to involve less invasive approaches. They were pioneered as early as 1806 by the invention of the Lichtleiter, considered to be the first documented prototype of what is now called an endoscope (Spaner et al. 1997). This device used to inspect the urethra, consisted of a slim tube connected to a chamber illuminated by a candle. A set of mirrors provided sight of the inside of the cavity while illuminating it. Numerous subsequent technical innovations allowed these devices to increase in versatility towards the endoscope we know today. Most notably, the development of bending or flexible endoscope sections in the first part of the 20th century and later, in the 70s, the development and miniaturization of the charge-coupling device (CCD) technologies led to the first flexible endoscope with an embedded camera in 1983 (Wheeler 1986). Since then, the field of flexible endoscopy has evolved from an explorative to a surgical technique with the additions of channels allowing the surgeons to insert tools in the endoscope to perform biopsies, dissection, and various other surgical acts.

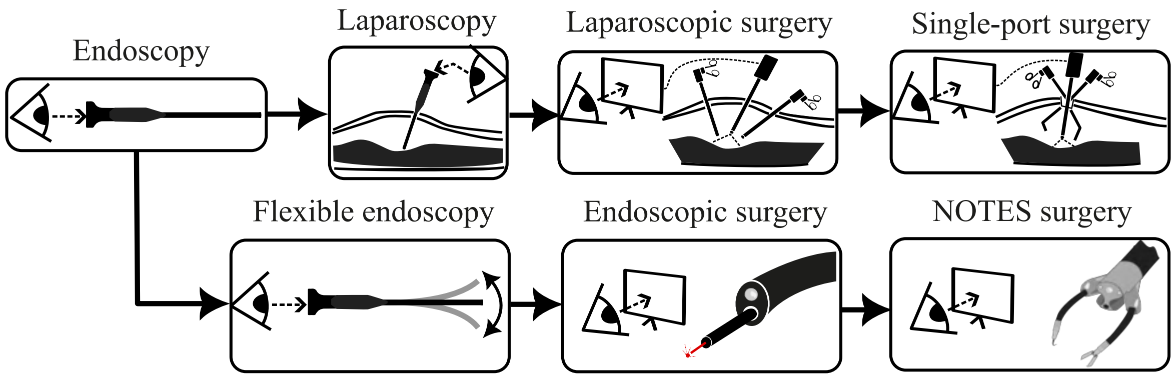


Figure 1.1: Illustration of the evolution of endoscopic techniques.

As illustrated in Figure 1.1, another major surgical innovation was developed in parallel to flexible endoscopy: the laparoscopy, a technique used to access the internal cavities, usually in the abdomen, through small incisions at the surface. It was initially used for exploration as early as 1901, with a rudimentary apparatus consisting of a slender hollow tube allowing the physician to observe the inside of an inflated abdomen (Peter et al. 2008). Progress in rigid endoscopy throughout the 20th century, including the distal integration of a CCD sensor, led to more sophisticated laparoscopes and a shift from exploration to surgical procedures, with a first laparoscopic surgical procedure – a cholecystectomy – in 1987 (Peter et al. 2008). With the potential to be used for numerous types of surgeries, this innovation would quickly become a routine technique; millions of laparascopic surgeries are now performed worldwide each year.

Endoscopic surgeries are considered to be minimally invasive surgery (MIS), due to the reduced patient trauma and minimized scaring. Beyond the obvious aesthetic improvements, MIS significantly reduces overall patient trauma, leading to faster recovery times, reduced post-operative pain, and shorter hospital stays (Darzi et al. 2004). We also saw in recent years the rise of new MIS techniques, including single-port procedures that consist in inserting multiple tools through a single entry point, usually located at the umbilic, and NOTES, i.e., Natural Orifice Translumenal Endoscopic Surgery (Bardaro et al. 2006). NOTES procedures consist in reaching the surgical site through natural orifices (i.e., mouth, anus, or vagina) by making a small internal incision, for instance in the stomach wall, through which endoscopic tools are inserted. This technique also motivated the development of more complex flexible endoscopic systems with multiple distal tools and triangulated configuration of these tools (Swanström et al. 2008).

However, in spite of a clear advantage for the patient, MIS techniques are usually less convenient for the surgeons who have to perform long surgeries in awkward postures (see Figure 1.2 for the case of laparoscopy), monitor the surgical site through a screen, and lose the sense of direct touch (Schostek et al. 2009). Thus, a robotization of these medical devices was proposed to allow a single surgeon to perform the complex manipulation of the devices while increasing the comfort and making the control of MIS tools more intuitive.

Figure 1.2: Laparoscopic surgery performed manually[1].

1.1.2 Robot-assisted minimally invasive surgery

Robot assistance for laparoscopy

One of the first uses of robots in the operating rooms was to hold the laparoscope (equipped with the endoscopic camera) that had until then to be either held by an assistant, sometimes for hours, or rigidly attached to the operating table using a passive endoscope holder. The first commercially available robotic endoscope holder was the Automated Endoscopic System for Optimal Positioning (AESOP® by Computer Motion Inc.), a voice controlled robot assistant for the operating room that is now discontinued. Subsequent systems introduced a robotization of the tools, allowing a single surgeon to remotely perform the surgery from a master console. An iconic example of these robots is the da Vinci® laparoscopic platform by Intuitive Surgical whose four arms can be teleoperated from a master console (see Figure 1.3a) using 3D visual feedback.

[1]Source: annistongeneralsurgery.com

Although the da Vinci robot had a monopoly in the last decade, other robotic platforms for laparoscopy are now entering the market, for instance Versius® (CMR surgical), Ottava (Johnson & Johnson), or Hugo™ (Medtronic).

(a) (b)

Figure 1.3: Intuitive Surgical da Vinci robot Si for robot-assisted laparoscopy. The surgeon remotely controls the robot (b) from the master console (a). Source: Pugin et al. (2011).

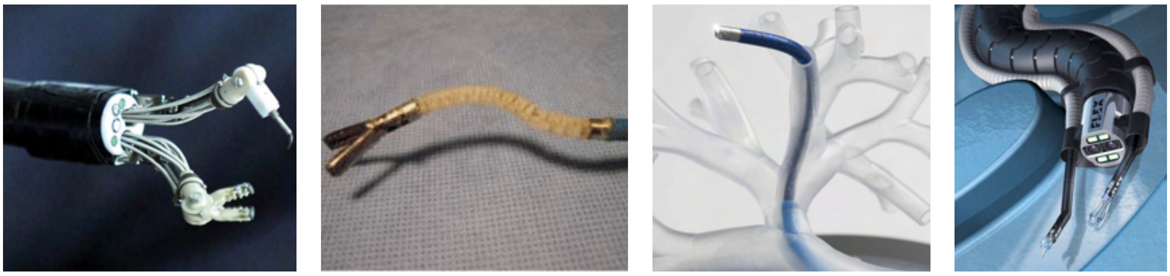
Robotic assistance for flexible endoscopy

Flexible endoscopy is also an interesting target for robotization due to the complex mapping between proximal actuation and distal motions, typically the bending of endoscope in one or two planes. The manipulation of classical flexible endoscopes (e.g., gastroscopes) is not intuitive since both bending are controlled by coaxial knobs. Robotization of the endoscopes enables a more intuitive control, for instance using a joystick (Lee et al. 2019) or an haptic interface (Reilink et al. 2011). Moreover, complex devices such as the Anubiscope (Dallemagne et al. 2010, see Figure 1.4a) were developed to perform NOTES and, more generally, endoluminal or transluminal surgeries. The manipulation of the endoscope and the two endoscopic tools inserted in the channels (10 DOF in total) require two or more highly trained surgeons to coordinate their actions in order to perform a surgery (see Figure 1.4b). Robot assistance was then naturally proposed to simplify the control of such systems. In the case of the aforementioned Anubiscope, the STRAS robotic platform (Zorn et al. 2018, see Figure 1.4c) allows a single surgeon to control the 10 DOF of the system from a master interface. Similarly, the MASTER (Phee et al. 2010), ViaCath (Abbott et al. 2007), or Flex™ (Robotics Surgical) systems are designed to provide a dexterous and intuitive control of complex endoscopic tools (see Figure 1.5).

Robotic-assistance is also impactful or even necessary for single-port laparoscopic surgery. In single-port surgery, all the surgical tools are inserted through a single entry-point such that their manipulation is complex due to the limited view and cluttered space causing blockage or tools collisions (Kaouk et al. 2009). Therefore, complex

(a) (b) (c)

Figure 1.4: (a) Anubiscope endoscopic system (Dallemagne et al. 2010). (b) NOTES surgery with the Anubiscope (Nageotte et al. 2020). (c) Using a robotic platform (STRAS, Zorn et al. 2018), a single surgeon can control the 10 DOF of the Anubiscope.



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

Figure 1.5: Robotized platform for endoluminal and transluminal surgeries. (a) MASTER (Phee et al. 2012). (b) ViaCath (Abbott et al. 2007). (c) MONARCH™ (Auris Health). (d) Flex™ (Robotics Surgical).

robotized tools were introduced to increase dexterity, including curved, flexible, or articulated tools (Nelson et al. 2017). An example of recent single-port surgical robot is the Intuitive Surgical SP (see Figure 1.6).

1.1.3 Autonomy levels in robot-assisted surgery

Regardless of their final application, robotic technologies for surgical assistance aim first to provide dexterous and ergonomic control over surgical tools. To date, these robots have been endowed with little to no autonomy and are designed to precisely replicate the motions performed by the surgeon on the master console. This direct telemanipulation scheme is typical of surgical robots used in laparoscopy and endoscopic surgery in general. However, there is a trend towards an increased robot autonomy level to reduce the cognitive load on the surgeon (Ciuti et al. 2020; Mayor et al. 2022).

The question of autonomy level is central in medical robotics as, besides the purely technical considerations, ethical and regulatory concerns naturally limit the extent to which such systems can make decisions on their own. Surgical robots can be characterized by their degree of autonomy, ranging from none at all, such that the surgeon is in charge of every decision and initiates every action, to total autonomy, such that the robot

Figure 1.6: Distal end of the Intuitive Surgical SP robot for single-port surgery.

can operate without human intervention or supervision. Different subdivisions of the autonomy levels have been proposed with, for example, six levels (Yang et al. 2017) or four levels (Yip et al. 2017), but there is a consensus on the six levels of autonomy proposed by Yang et al. (2017) in recent literature (Haidegger 2019; Attanasio et al. 2021). Regardless of the exact subdivision used, the general stages of automation are as follows:

- **No autonomy** – the surgeon initiates and executes every actions; automatic behaviors are limited to basic control, including motion scaling, master-follower DOF mapping, or tremor filtering;
- **Assistance** – although the surgeon retains full control, the system provides assistance that can either be passive (e.g., augmented reality) or active as in the case of haptic feedback or guidance (Attanasio et al. 2021);
- **Partial autonomy** – part of the procedure is autonomously performed by the robot, but under supervision of the surgeon. It is the subdivision of this level of autonomy that, based on the level of decision making entrusted to the robot, differs between classifications. The consensus is to further divide this autonomy level in three: task autonomy, conditional autonomy, and high autonomy (Yang et al. 2017). We will keep using the term **partial autonomy** as this level of detail is sufficient in the following sections;
- **Full autonomy** – the robot can autonomously perform the surgery, with no need for human supervision.

No surgical robot can yet be qualified as fully autonomous and commercial systems with **partial autonomy** are typically only used when the environment and the task to perform are perfectly defined. This is the case, for example, of the CyberKnife system (Accuray) used to perform robot-assisted radiotherapy – although, not strictly surgical. The robot trajectory is planned from preoperative data (reconstructed 3D images) and subsequently executed automatically using intra-operative registration from real-time X-ray imaging. The high level of automation is made possible by the fact that targets are usually fixed relative to rigid structures (e.g., skull or spine) or, if the former is not true, located using fiducials visible on X-ray images (Kilby et al. 2010). Ongoing research aims at providing **partial autonomy** in the form of sub-task automation for applications such as automatic suturing, knot tying, or needle insertion. However, to date, most of the robots clinically used for MIS have **no autonomy**: these robots are designed to provide an accurate teleoperation such that the robot only executes the motions performed by the surgeon at the master side.

Some commercial robots are at the so-called **assistance** level, characterized by the fact that although the system performs autonomous actions, the surgeon remains in full control of the procedure. Commercial products include assistive robots for orthopedic surgery such as the ROSA Knee robot (Zimmer Biomet), which automatically positions a physical guide for cutting or drilling bone. Similarly, the MAKO robot (Stryker) provides tools (e.g., drill or saw) attached to a robotic arm that can constrain the surgeon motion (e.g., along a straight line) or prevent dangerous actions through the application of forces. Both of these systems are manipulated in a way such that the robot and the surgeon physically share the operating room and operate side by side as opposed to teleoperated systems, which are controlled remotely. Various challenges have limited the spread of robotic assistance based on automatic execution modes mostly to fields such as orthopedic or neurosurgery, where the rigid environment simplifies the planning and registration of the pre-operative plan. Even if most of these techniques, regardless of their level of autonomy, have not yet reached clinical use in the field of MIS, they have been the subject of a significant body of research over the last two decades. In the following section, a brief overview of the literature on assistive behaviors in the field of MIS is provided.

1.2 Assistive robotic behaviors for MIS

A so-called assistive behavior aims at reducing the cognitive workload of the surgeon by relying on automation to perform low-level tasks (e.g., filtering or control aspects), repetitive actions (e.g., scanning), or other aspects of the task for which an automatic mode is more suited. This allows the surgeon to take advantage of the capability of the robotic system, for instance the presence of sensors, to improve the overall performance of the surgery, including accuracy, execution time, or repeatability. Ideally, the automation will control aspects of the task that require little to no decision making, leaving high-level control to the surgeon. In the following, assistive behaviors are presented, following their level of autonomy.

1.2.1 Assisted robot control

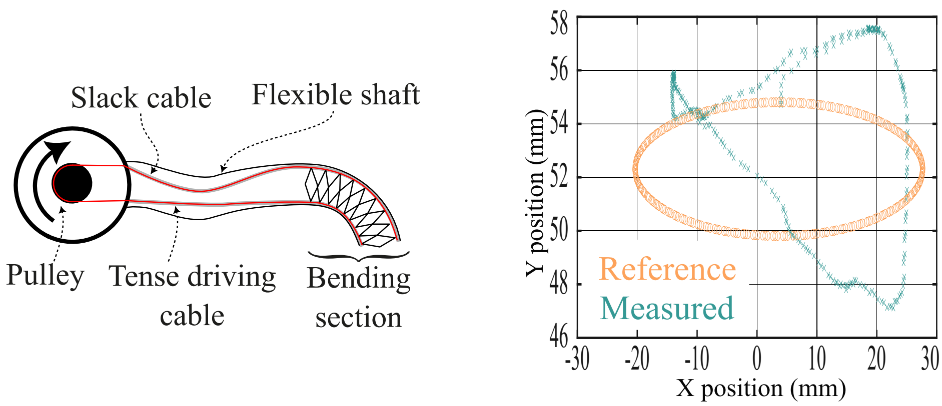
This type of assistance has **no autonomy**. It consists in control algorithms augmenting or improving robot control by the operator.

Scaling and tremor suppression

The majority of surgical robotic systems, especially for MIS, are teleoperated such that the motions performed by the surgeon are first processed by the system. This allows the implementation of control schemes designed to enhance the accuracy of the surgeon. In particular, position scaling can be used to map master displacements into smaller displacements of the follower robot, hence reducing the inaccuracies introduced by the surgeon on the master side (Prasad et al. 2004). Filtering techniques were also used to reduce tremors, both for teleoperated robots (e.g., implemented on da Vinci robots, Leal Ghezzi et al. 2016) and for hand-held robots (Taylor et al. 1999).

Backlash compensation

Due to the need for miniaturization, many robotic systems for MIS are actuated from the proximal side using tendon-sheath mechanisms (TSM) or other cable actuation technologies that suffer from mechanical backlash in the transmission. Backlash in TSM is caused by the necessary slack in the antagonist cables (see Figure 1.7a), their elasticity, and the friction between the cable and its sheath (Do et al. 2014). This is especially the case with flexible endoscopes, for which the effects are amplified by their long actuation path that makes backlash effects especially strong. Backlash introduces hysteresis-like effects in the transmission that are experienced as control latencies by the surgeon. It was demonstrated that an operator can cope with small backlash in the transmission during telemanipulated robot-assisted MIS (Kim et al. 2020), but this backlash tends to degrade performance and increase mental load (Peine et al. 2012). Furthermore, backlash also impairs robot positioning, with significant tracking errors in the operational space as visible in Figure 1.7b. Such inaccuracies have to be corrected to implement advanced robot behaviors such as shared control or physiological motion compensation (see Section 1.2.2).



(a) (b)

Figure 1.7: (a) Illustration of cable actuated bending, adapted from (Bardou et al. 2012). (b) If not compensated, backlash in the transmission results in significant tracking errors in operational space, here with a flexible endoscopic tool of the STRAS platform (Aleluia Porto 2021).

Then, regardless of the system level of autonomy, appropriate control strategies have to be implemented in order to compensate for backlash. Approaches for backlash compensation are typically based on a model and implemented as a feedforward controller that directly compensates the apparent deadzone (i.e., backlash width at a given position) by model inversion (Bardou et al. 2012; Aleluia Porto et al. 2019). Alternatively, one can use a switching controller that increases actuator velocity when said actuator crosses a deadzone, hence reducing the backlash experienced by an operator (Reilink et al. 2013b).

1.2.2 Shared control paradigms

Another approach especially pertinent to MIS is shared control, an intermediary autonomy level where a human operator effectively shares the control with the automation (Yip et al. 2017; Niemeyer et al. 2016) and that would fall within the **assistance** level described by Yang et al. (2017).

A brief taxonomy

Shared control (SC) is a general term referring to control paradigms where the operator shares the control with automation. The division can sometimes be such that the operator and automation control distinct DOF of the robot, which can then be seen as a form of semi-autonomous control since the automation has full control over a partition of the DOF. For instance, in a bi-manual robotic setup for laparascopic surgery, the operator could control the tools while the robot arm holding the camera is automatically controlled (Weede et al. 2011). However, in many cases, the control of the DOF of the system is shared such that both operator and automation simultaneously control the same DOF.

More generally, SC algorithms can be classified by the method used to combine the commands of the operator and automation. State shared control (SSC) consists in combining these commands at robot state level, typically through a weighted sum (Dragan et al. 2013). In the literature, this type of SC is also referred to as "input-mixing SC" (Katzourakis et al. 2011), "mixed initiative control" (Saeidi et al. 2017), or "policy blending" (Dragan et al. 2013). Alternatively, the commands can be combined through a physical interaction. More precisely, both the operator and the automation apply forces on the master interface and the resulting velocity or position is directly used as a command (see Figure 1.8). In the case of collaborative robotics, the same applies except that forces applied by the operator and the automation directly affect the state of the robot. This type of SC is referred to as haptic shared control (HSC, Abbink et al. 2012).

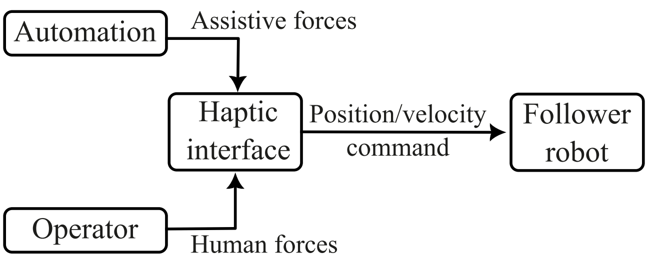


Figure 1.8: Simplified illustration of the general HSC concept for telemanipulation, see Abbink et al. (2012) for a more detailed representation.

A major difference between these two schemes lies in how the arbitration between operator and automation is handled or, in other words, how authority is allocated (Abbink et al. 2018). HSC provides an effective way to arbitrate between operator and automation. Limiting the maximal force that can be applied by the automation effectively ensures that the operator will always be able to take back control if needed. Additionally, the haptic interaction clearly and continuously communicates the intentions of the automation to the operator (Abbink et al. 2012). On the contrary, SSC does not provide a continuous feedback of the system and the operator does not necessarily have the final control over task execution. Although this is generally not desirable in surgical scenarios, the complexity or high dynamics of a task can justify the use of SSC strategies, for instance, in the case of physiological motion compensation. Applications of SSC and HSC to robot-assisted MIS are presented in the following sections.

State shared control

SSC is generally used to combine the actions of the operator with an automatic prediction of what is intended. As this is done at the state level, no physical interaction with the operator is required such that other communication chanels can be used. For instance, outside the context of surgical robotics, SSC has been used to assist robot control from motion capture (Dragan et al. 2013; Javdani et al. 2018).

In the field of robot-assisted MIS, SSC is predominately used to stabilize the robot with respect to tissues moving because of breathing, heartbeat, or other physiological motions. SSC can be used to track the displacements and compensate for them such that the surgeon operates guided by the static display of the environment. This approach was for instance applied to minimally invasive cardiac surgery, where the surface of the beating heart can be tracked in endoscopic images (Ortmaier et al. 2005; Richa et al. 2010). Respiratory motions can be compensated during robot-assisted needle insertion using real-time intra-operative imaging, for instance X-ray (Baksic et al. 2021). Similarly, an approach for respiratory motions compensation was proposed to assist the manipulation of a robotized flexible endoscope (Ott et al. 2011). Besides motion compensation, SSC was for instance used during contactless sub-tasks (e.g., large motion from A to B) such that the robot motion was the super-imposition of automatic and operator motions (Padoy et al. 2011). The automatic trajectory was learned beforehand from demonstrations and its execution triggered by gesture recognition.

Haptic shared control

The first use of HSC consisted in implementing so-called virtual fixtures (VF) to physically constrain the operator's motion through the application of forces on the master interface (Rosenberg 1993). Intuitively, VF are used to generate virtual interactions that the operator can feel through forces applied on a force-enabled master interface, hence exploiting haptic communication rather than relying on visual or auditive cues. A survey by Bowyer et al. (2014) provides a comprehensive overview of VF methods, which can be broadly divided in two categories. First, VF can be used to prevent forbidden (e.g., dangerous) motions by virtually restraining the operator inside or outside a given region. In robot-assisted MIS, this approach is typically used to protect critical tissues such as veins, arteries, or nerves. Dynamic forbidden region VF were also introduced to cope with moving environnements such that the VF effectively tracks the motion of the tissue (Gibo et al. 2009; Marinho et al. 2019), with applications such as beating heart surgery (Ren et al. 2008; Rydén et al. 2012).

A second type of VF, usually referred to as guidance VF, consists in applying continuous forces to guide the motion along a path or a surface. This type of VF is best described by the more general HSC framework introduced by Abbink et al. (2012) previously discussed, since operator and automation simultaneously apply forces to the master robot to achieve a same task. Haptic guidance schemes can be used to assist path following tasks during contact-free (Xiong et al. 2017; Olivieri et al. 2018) or contact-rich (e.g., dissection tasks, Moccia et al. 2019; Feng et al. 2021) trajectories. Similarly, planar haptic guidance was used to assist knot tying (Chen et al. 2016) and Selvaggio et al. (2019) proposed a HSC method to guide the surgeon towards optimal needle grasping poses to assist robot-assisted suturing. HSC was also used to physically guide the surgeon towards the center of the lumen during the insertion of a teleoperated flexible endoscope (Reilink et al. 2011).

1.2.3 Automatic sub-task execution

It is usually preferable to keep the surgeon in-the-loop during robot-assisted surgery for safety and responsibility reasons, but the automation of some aspects of the surgery remains interesting in many scenarios. This is especially the case for repetitive gestures that are complex to execute or require high dexterity, but little to no decision-making. For instance, the automation of suturing or knot tying sub-tasks has been extensively investigated in the literature (Fontanelli et al. 2018). The proposed approaches range from geometrical planning of automatic trajectories (Nageotte et al. 2009; Liu et al. 2016) to the use of machine learning to learn the skills from human surgeons (Schulman et al. 2016; Osa et al. 2018). Methods were also proposed to perform automatic scanning using ultrasound (US) probes (Zhan et al. 2020) or optical coherence tomography (OCT) devices (Zhang et al. 2021). Similarly, tissue stiffness mapping from force sensing is a likely candidate for automation, for instance to subsequently provide haptic guidance based on the generated tissue topography (Wang et al. 2017). Although far from exhaustive, this list illustrates the range of tasks or sub-tasks for which a higher level of automation would be of interest during surgery.

1.3 Challenges to assistive robotic behaviors for MIS

Of all the assistive robotic behaviors previously introduced, very few are currently implemented outside controlled environments, i.e., research testbeds. Although that could be explained by regulatory and technological maturity issues, there are inherent challenges in MIS that make automation very difficult. As discussed in the following, these challenges include organ motions and deformations, limited intra-operative measurements, and robot positioning inaccuracies.

1.3.1 Intra-operative task registrations

A typical robot-assisted surgical procedure starts with the acquisition of preoperative patient-specific data, e.g., using magnetic resonance imaging (MRI) or computed tomography (CT), which is used to plan the surgical task. An additional step consists then in the registration of the corresponding path or trajectory into the operational space of the robot so that the task can be executed. Note that if the robot has no autonomy at all (e.g., in telerobotics), then the surgeon performs this registration using available visual feedback and knowledge of the anatomy. In any other scenario, whether the task has to be automatically executed by the robot or exploited to assist the surgeon, the system will have to perform this registration automatically. This is typically done by matching intra-operative measurements of fiducials or other recognizable structures with some acquired from the preoperative data initially used to plan the task (Troccaz 2013, Chapter 3).

In the case of endoscopic robot-assisted surgery, the camera images or other real-time medical images (e.g., US or OCT) are usually the only available source of intra-operative information, making registration of preoperative data especially challenging for the following reasons. First, the *in-situ* image measurements are 2D, whereas pre-operative images are typically 3D. A stereoscopic camera-based depth estimation would partially alleviate the problem, but many MIS robots are not equipped with them. Additional technical limitations of endoscopic images include intermittent measurement unavailability due to occlusions, smoke, or specular reflections. Furthermore, the tissues might be deformable such that their surface has to be tracked in real-time. All of the aforementioned issues limit the achievable accuracy of the *in-situ* registration of a task planned intra-operatively and, consequently, also limit the precision of its execution.

To make the challenges more explicit, let us assume that the task to perform can be defined as a trajectory located with respect to tissues of interests, for instance a tumor. To do so, this tumor is segmented in some pre-operative image and a desired robot tool trajectory is computed (e.g., for dissection or scanning). During the procedure, endoscopic images are used to track visual landmarks and reconstruct the environment, for instance through visual simultaneous localization and mapping, i.e., SLAM (Mahmoud et al. 2019). In the absence of tissue deformation, organ motion, or image artifacts, the registration error might be reasonable, e.g., 1 to 2 mm surface reconstruction RMSE (root mean square error) for state-of-the-art SLAM methods (Mahmoud et al. 2019), but it could also be much larger. This tissue registration is used to express the previously planned trajectory in the reference frame of the robot. In the following, this registered trajectory will be referred to as the planned trajectory; it is the one that would, in the absence of robot positioning inaccuracies, be executed by the robot. Due to registration errors, it can be different from the trajectory actually preferred by the surgeon as illustrated in Figure 1.9a. It should be noted that if the trajectory is directly planned *in-situ*, from intra-operative endoscopic images, then the previously discussed issues impact also the planned trajectory. For example, the surgeon could plan a trajectory directly in the endoscopic image, but subsequent tissue tracking or depth-estimation errors can introduce registration inaccuracies (see Figure 1.9b).

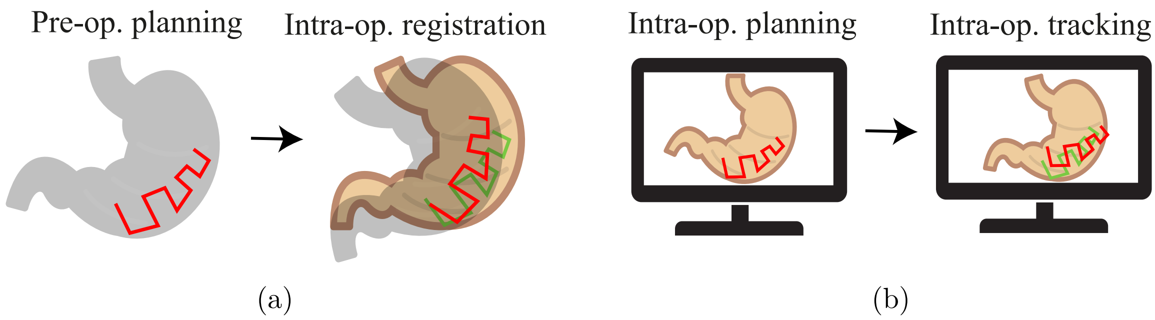


Figure 1.9: Illustration of *in-situ* task registration errors arising from preoperative to intra-operative registration (a) or errors introduced over time due to tissue tracking inaccuracies (b).

1.3.2 Inaccurate robot modeling

Once a trajectory is planned, it has to be executed by the robot. Medical robots typically lack task space sensors (i.e., distal) such that position control is often performed in open-loop from kinematic models. Then, even assuming a correct hand-eye registration allowing to perform the registration from camera to robot reference frames, positioning errors can be introduced by inaccurate robot models. In the case of a robot with actuation backlash, these errors are potentially very significant (see Section 1.2.1). Furthermore, if the robot has to execute the task using a tool attached *in-situ* to its end-effector (e.g., a needle seized to perform suturing), then an additional hand-tool registration error might appear.

1.3.3 Effects of task and robot model inaccuracies

The task registration and robot positioning issues discussed in the previous sections can be seen as modeling errors on:

- the model of the task, typically in the form of a trajectory expressed in the robot frame of reference;
- the kinematic model of the robot and its transmission that can include a backlash model for many medical robots as previously discussed.

Together, they introduce inaccuracies during automatic task execution as can be expected, but also adversely impact the performance of shared control strategies.

Effects on automatic task execution

First, let us consider the case of a task planned intra-operatively and executed automatically by a surgical robot. Once registered in the robot frame of reference, the planned trajectory is the reference sent to robot the controller. Due to robot modeling errors, the trajectory actually executed might differ from the planned trajectory. However, in the presence of task modeling (e.g., registration) errors, the planned trajectory sent as a reference to the controller is itself different from the one actually preferred by the surgeon. As illustrated in Figure 1.10, a surgeon supervising the task execution (e.g., from visual feedback) might then not be satisfied by the executed trajectory.

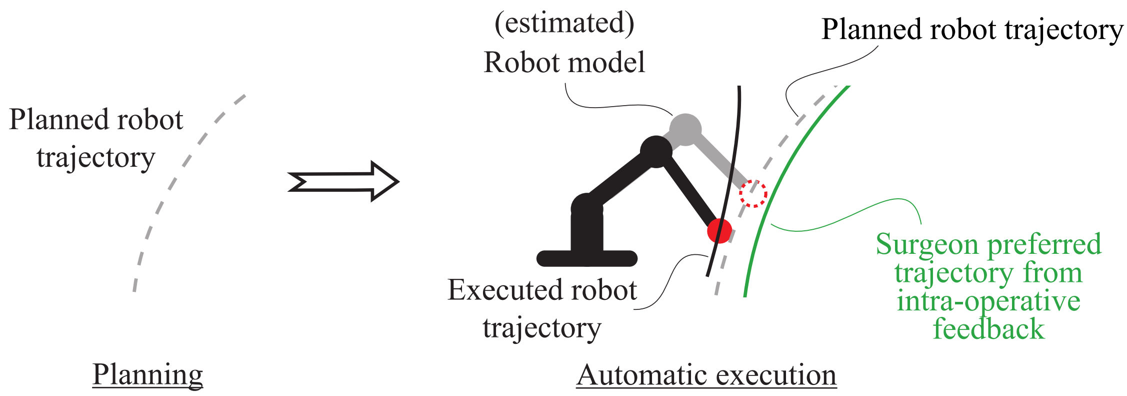


Figure 1.10: Illustration of an inaccurate automatic task execution resulting from the task and robot modeling errors.

Effects on shared control

Challenges to robot-assisted MIS were so far presented with the point of view of automatic task execution. However, the presence of the surgeon in-the-loop does alleviate the need for accurate modeling, as this person will still be able to perform the task using the available visual feedback. This advocates for shared-control as a way to use the automation while leaving the operator in charge of coping with modeling inaccuracies.

As the surgeon should maintain as much control as possible, HSC should be preferred when possible such that the surgeon gets continuous feedback about the intention of the assistive system – and can override it if necessary. Haptic guidance, a form of HSC that guides the motion of the operator through the application of forces, has been shown to improve task execution accuracy and reduce cognitive load in telerobotics (Enayati et al. 2016). However, although final control remains with the surgeon, the robot still needs a correct representation of its environment to provide useful assistance. The required models depend on the nature of the shared control scheme considered, but they typically include one or several of the following list: a model of the environment, a task model in the form of a trajectory, the registration of said task in the operational space of the robot, and the kinematic model of the robot.

In the presence of such modeling errors, the assistive behaviors implemented will likewise be inaccurate. For instance, in the case of HSC approaches, the surgeon would be physically guided towards a trajectory that is not the preferred one, which can be disturbing and most likely unhelpful. A study carried out by Oosterhout et al. (2015) demonstrated that inaccurate haptic guidance degrades performance. Other studies also point towards the same conclusion (Smisek et al. 2015; Lee et al. 2017), which would advocate for an *in-situ* correction of the models to reduce conflicts and improve performance. Therefore, an ideal assistive HSC system should adapt its internal models online to improve the provided assistance.

It is important to note that these task modeling issues could sometimes be solved using exteroceptive measurements to improve the haptic guidance online. For instance, the path of VF has been updated online from scene reconstruction using endoscopic images to assist polyp dissection (Moccia et al. 2019) or LASER surgery (Olivieri et al. 2018). Force sensing can also be used to map the environment (topology and stiffness) through palpation, which in turn can be used to update task models for the generation of VF (Wang et al. 2017; Yasin et al. 2021). However, such approaches exploit sensor information that could be unavailable or inaccurate, especially visual information as previously discussed. Furthermore, sensor-based approaches ignore the fact that the operator might be in disagreement with the generated haptic guidance because of surgery circumstances.

1.4 Content of the thesis

1.4.1 Organization of the manuscript

In this thesis, we explore several ways to refine task and robot models *in-situ*, with a focus on online task registration and backlash identification, both critical in many MIS scenarios as previously discussed.

In the first part of the document, we propose new methods for the *in-situ* identification of backlash in flexible endoscopic robots. The work is organized as follows.

Chapter 2 After introducing the scientific literature on flexible endoscope backlash modeling and estimation, we present an approach for online and *in-situ* backlash learning adapted to the case of flexible endoscopy. To do so, we build on the notion of discontinuous Kalman filtering to propose a unified learning method for multi-DOF backlash estimation. The proposed approach is validated through comprehensive simulations of a flexible endoscopic robot.

Chapter 3 Existing methods for automatic backlash width estimation and compensation most often consider eye-to-hand endoscopes. In this chapter, we propose an original problem formulation inspired by the concepts of SLAM to perform the *in-situ* backlash identification of eye-in-hand endoscopes from endoscopic images only. The learning algorithm presented in Chapter 2 is then used to implement the learning method, which is validated both in simulation and on a real endoscopic robot.

Chapter 4 Finally, we explore how degraded image measurements that are insufficient to reconstruct the pose of the robot can still be used to extract information about the backlash model. We present a backlash width estimation method that uses discrete motion detection events to learn the backlash width model, with no need for robot pose estimation. We also demonstrate through experimental results that the proposed approach can be combined with more traditional robot pose estimation methods to reconstruct the complete backlash model.

In part 1, the proposed methods use traditional sensor-based approaches whereas in Part II, it is the presence of the operator in-the-loop that is exploited. This second approach does not rely on exteroceptive sensors that could become unavailable or erroneous, but rather uses the operator as an information source. In the second part of the document, we explore adaptive SC schemes to correct task and robot models online, while these models are simultaneously used to assist the operator. As it is most relevant to surgical applications, we will consider haptic guidance throughout Part II, but our work could apply to other HSC or SSC paradigms. The work is organized as follows.

Chapter 5 After introducing the state-of-the-Art methods for adaptive HSC and human-robot interaction, we derive the problem formulation for online task and robot models learning. Then, we propose an optimization-based approach to refine these models using the operator in-the-loop as an information source. The proposed algorithm is illustrated on simulated toy scenarios.

Chapter 6 The method introduced in Chapter 5 is experimentally evaluated on generic teleoperation tasks. Relevant implementation details are discussed, including the question of parameters identifiability and hyperparameters tuning. Then, comprehensive experimental results are reported, demonstrating the performance of the proposed method on a remote drawing task.

Chapter 7 In this chapter, the effectiveness of the proposed adaptive haptic guidance method on human operators is evaluated in a user study. Furthermore, practical questions are addressed, including the effect of so-called “workspace clutching” on the performance of haptic guidance.

Finally, we conclude and present potential research avenues for our work, including how multimodal learning could be used to combine sensor information with the information extracted from the operator in-the-loop. Preliminary work on the simultaneous correction of backlash and task registration is also presented, using concepts from chapters 2 and 5.

1.4.2 Contributions and scientific communications

The main contributions of this thesis and associated publications are listed below (a more detailled list can be found in Appendix A).

Contributions to the image-based online identification of endoscopic robot models:

- we build on discontinuous Kalman filtering to propose an unified and general framework for multi-DOF backlash estimation adapted to flexible endoscopy online model identification. The method and results are presented in Chapter 2 and were accepted for publication (Poignonec et al. 2023a);
- we propose a novel approach for image-based *in-situ* backlash estimation suited to eye-in-hand endoscopes and validate this method experimentally on a clinical

endoscope. The method and results are presented in Chapter 3 and were submitted for publication (Poignonec et al. 2023b);

- we propose a novel approach for *in-situ* backlash estimation that is based on image-based motion detection and that therefore does not require robot pose estimation. The method and results are presented in Chapter 4 and a journal publication is ongoing.

Contributions to the online task and robot models from user inputs:

- we propose a new method for simultaneous task and robot model learning. This is a unified framework that allows the online correction of model parameters from the observation of operator actions. We present a recursive implementation based on Kalman filtering and thorough simulation and experimental validation. The theoretical developments are presented in Chapter 5 and the experimental results in a telerobotic context in Chapter 6. Journal publication is ongoing;
- As an alternative to Kalman filtering, we propose a novel approach for online task registration from the observation of operator actions based on adaptive size sliding window optimization. The method, including its application to adaptive haptic guidance, and experimental results with an operator in-the-loop are partially presented in Appendix E and discussed in the closing chapter of Part II. This material was published in (Poignonec et al. 2021);
- we present comprehensive results concerning the effect of adaptive guidance on operator performance and perceived arduousness collected during a user study. The user study design and main results are presented in Chapter 7. Journal publication is ongoing.

Part I

In-situ backlash estimation for robotized flexible endoscopy

Chapter 2

A Bayesian approach for online endoscopic robots model identification

Contents

2.1 Related work 20
  2.1.1 Structure of cable-driven flexible endoscopes 20
  2.1.2 Backlash modeling and estimation 21
  2.1.3 *In-situ* backlash estimation 22
2.2 Proposed approach 23
  2.2.1 Robot kinematic modeling 24
  2.2.2 State and parameters estimation 25
2.3 One DOF illustration 28
  2.3.1 One-dimensional simulation 28
  2.3.2 Comparison with classical EKF 30
2.4 Simulation results 31
  2.4.1 Simulation setup 31
  2.4.2 Results 33
2.5 Conclusions 33

Backlash is a common issue for the accurate control of medical robots and especially cable-actuated flexible endoscopes. Accurate backlash estimation and modeling can be used to improve the control of robotic flexible endoscopes, thus leading to improved user experience in telemanipulation, or to enable automatic or semi-automatic sub-tasks. As presented in the following section, there is a vast literature concerning the identification of cable actuation backlash models. Existing solutions can be divided between off-line and *in-situ* approaches, the former allowing a more precise modeling, but the later a greater robustness to modifications of the hysteretic behavior that often occurs during the insertion or manipulation of the endoscope. In this chapter, we propose an approach to perform backlash estimation *in-situ* while maintaining a model complexity greater than that of existing *in-situ* approaches, allowing to capture the shape of the hysteresis more accurately.

2.1 Related work

2.1.1 Structure of cable-driven flexible endoscopes

Cable-actuated flexible endoscopes are typically composed of a passive flexible section and an actuated bending section at the distal end (see Figure 2.1). Depending on the system, the distal end is also equipped with an endoscopic camera or with a surgical tool (e.g., gripper). There is generally one or two actuated bending DOF, such that the distal section can bend in one or two planes, respectively. The cable actuation is motorized at the proximal end such that each DOF is driven by a pulley as was illustrated in Figure 1.7a: the pulley, when actuated, changes the relative length of the two antagonist cables (i.e., Bowden cables, see Figure 2.1), which in turn causes the bending of the endoscope.

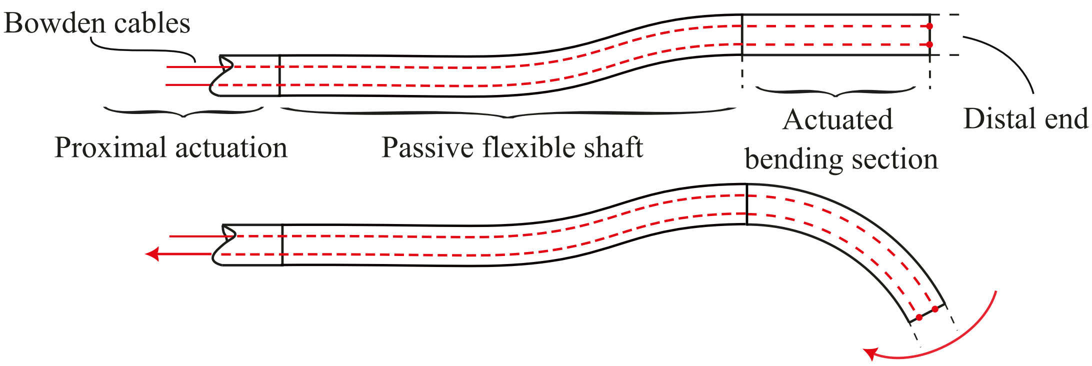


Figure 2.1: Illustration of cable-driven bending in one plane (i.e., 1 bending DOF).

Other DOF can be actuated, typically the orientation of the endoscope or its translation. For instance, the endoscope structure that will be considered in this chapter is a robotized endoscopic tool inserted in the channel of a larger endoscope. The endoscopic tool has three actuated DOF: the bending, the orientation within the channel, and the translation along the channel. It should be noted that although backlash is particularly present in the cable-actuated bending DOF, the rotation of the endoscope also exhibits a hysteretic behavior caused by the deformation of the passive flexible shaft and the friction with the channel or environment (Aleluia Porto et al. 2019). For the same reasons, the translation of the endoscope can also suffer from backlash if robotized, although to a lesser degree.

2.1.2 Backlash modeling and estimation

Some work alleviate backlash identification by relying on closed-loop control of the endoscopic system, using embedded sensors such as shape sensing devices or endoscopic cameras. In such cases, the underlying backlash model is not identified, and its effect has to be repeatedly rejected by the closed loop, which can be slow and induce delays in task execution. Consequently, model-based backlash compensation based on feed forward strategies is usually preferred.

The modeling of backlash in cable actuation can focus on the underlying physical phenomena: the elongation of the cables and the friction in the conduits. For instance, Chiang et al. (2009) proposed to model the mechanics of the cables and to estimate the cable's elongation and distal pulling force from proximal force and position sensors. Similarly, Sun et al. (2014) estimate physical parameters such as friction and curvature to model the elongation of the cable. Such physical models were used for the compensation of backlash in cable-actuated flexible endoscope (Xu et al. 2017): after training the cable elongation model offline, the authors use it to compensate backlash online with a feed-forward controller. However, physical parameters such as cable/sheath friction coefficients are generally not known and can change over time. Furthermore, the measurement of cable tension is generally not available, even on the proximal side, such that precise mechanical modeling of the cable is challenging.

Approaches not relying on exact mechanical models, but directly on hysteretic relationships between actuators positions and effector pose have also been proposed. The mathematical parameters of the models can be estimated from off-line identification as proposed by Do et al. (2014). The authors estimate the parameters of a simplified Bouc-Wen model (see Figure 2.2c) using proximal and distal cable displacement measurements. Although not necessarily applied to the modeling of cable actuation, other methods

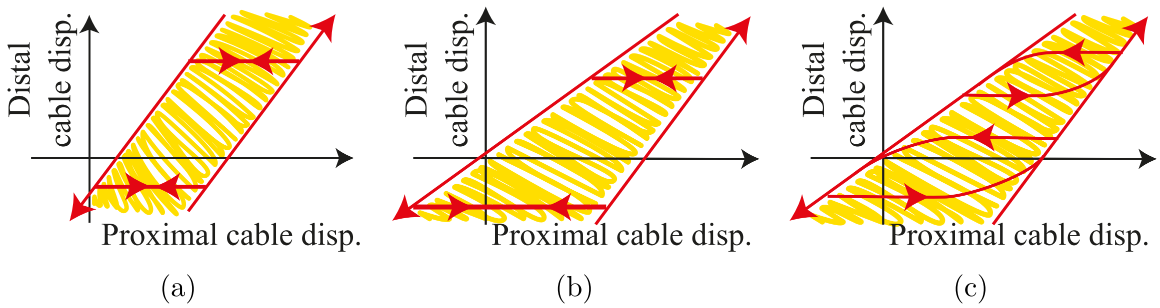
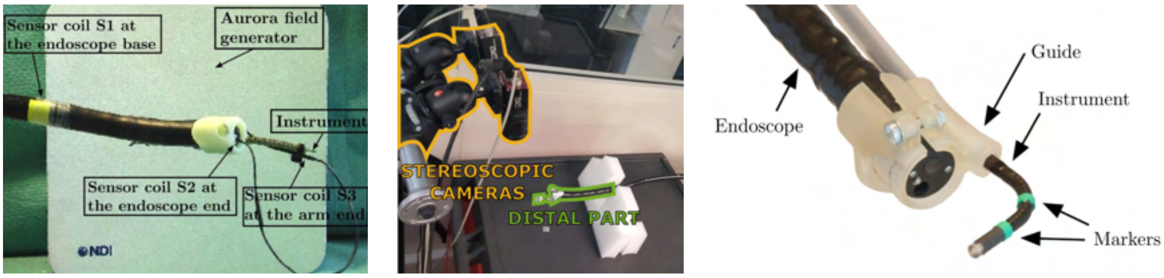


Figure 2.2: Illustration of most common analytical hysteresis models for parametric backlash modeling where the red arrows denote the direction of motion. (a) Linear branches and constant width. (b) Linear branches. (c) Asymmetric Bouc Wen model.

for parametric backlash model identification have been proposed. For instance, Vörös (2010) used iterative least square optimization to identify offline the parameters of a backlash model with linear branches (see Figure 2.2b).

Finally, non-parametric model identification can also be used to learn the inverse kinematic model linking distal endoscope configuration and proximal cable displacements, thus directly using the measured relations to implement the compensation scheme. Aleluia Porto et al. (2019) used neural networks to model the inverse kinematics of an endoscopic robot and Bardou et al. (2012) used a lookup table to record the backlash width, i.e., the deadzone crossed at each change of direction, for different actuator positions.

At this point, it should be noted that it is not possible to directly measure the distal cable displacements on endoscopic systems: only indirect measurements are available. External sensors are then used for offline backlash identification, for instance electromagnetic trackers (see Figure 2.3a) or stereoscopic cameras to track an optical marker (see Figure 2.3b). However, these solutions are not applicable to *in-situ* model identification (i.e., at the surgical site, using intra-operative sensors) and, consequently, neither are the identification methods previously presented. Of course, models identified offline can subsequently be used *in-situ* to implement backlash compensation, but the hysteretic relation between proximal and distal displacement is highly dependent on the shape (i.e., curvature) of the endoscope (Ott et al. 2011). The effect of shape on hysteretic behavior was also studied in the context of cable actuation for other applications, with similar conclusions (Dinh et al. 2016). Since the shape of the endoscope changes between the offline identification and the surgery due to the insertion of the endoscope at the beginning of the procedure, the models can potentially become inaccurate once *in-situ*. For instance, one can observe in Figure 2.4 that strong variation of the backlash model appears between subsequent acquisitions on the same robot. Therefore, *in situ* identification of backlash should be preferred.



(a) Magnetic tracker (b) Optical marker (c)

Figure 2.3: The endoscope end-effector pose can be obtained through external tracking (a-b) or *in-situ* pose estimation from endoscopic images (c). Sources (ordered list): Bardou et al. (2012), Aleluia Porto et al. (2019), Reilink et al. (2013b).

2.1.3 *In-situ* backlash estimation

Reilink et al. (2013b) used marker-based pose estimation from endoscopic images (see Figure 2.3c) to learn the parameters of a backlash model. The method consists in

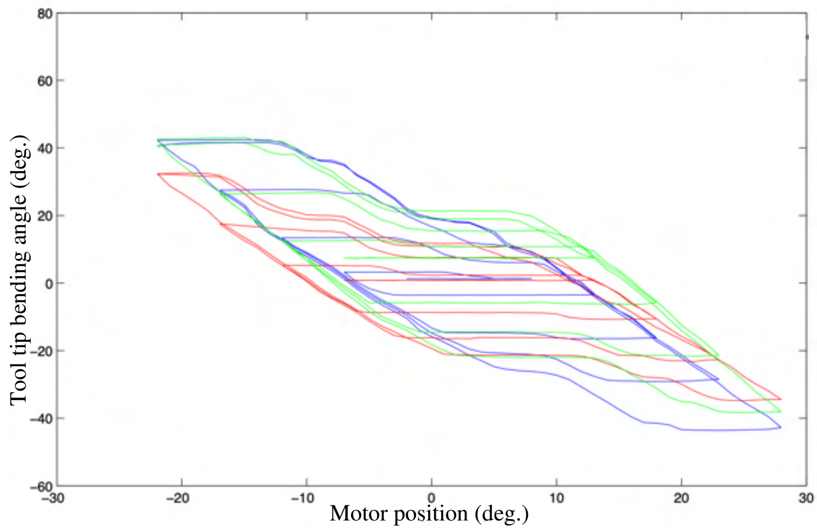


Figure 2.4: Hysteretic behaviors measured on a tendon driven DOF of a flexible endoscopic robot, between proximal actuator and distal configuration. The different colors correspond to different dates of measurement between *in vivo* (Poignonec et al. 2020).

continuously comparing actual motor positions with estimated ones and updating the model parameters to reduce the discrepancy. The backlash model has two parameters per DOF, as it assumes constant width and slope as depicted in Figure 2.2a. More recently, Baek et al. (2020) proposed a hybrid method relying on a model of backlash learned offline and an online pose estimation from endoscopic images. The authors used a Kalman filter to fuse the two joint angles estimations, resulting respectively from the backlash model and the online pose estimation, to improve the closed loop performance of the K-FLEX robotic system. However, the backlash model is not strictly learned *in-situ* and, besides, mostly used to replace pose estimation when it becomes unavailable due to occlusions.

2.2 Proposed approach

To the best of our knowledge, the approach proposed by Reilink et al. (2013b) is the only existing online method for *in-situ* multi-DOF backlash model identification. Its main limitation lies in the fact that two-parameter models are often unable to capture realistic backlash, and that variable width should be incorporated into the model. Another limitation of previous works is related to the fact that sensor noise is not taken into account, which may be critical as computer vision algorithms provides noisy estimations.

Therefore, we propose a generic online multi-DOF backlash learning method based on the concept of discontinuous EKF (DEKF, see Chatzis et al. 2017b). The proof of concept is demonstrated with a simulated scenario with realistic dimensions and model parameters. The approach has the advantage of being more general than the method proposed by Reilink et al. (2013b) as it allows to include additional backlash parameters. Moreover, by relying on a Bayesian formulation, the proposed approach allows to take into account measurements noise and to explicitly incorporate correlations. This last point will later be exploited in Chapter 3 to implement a SLAM-like algorithm, further demonstrating the versatility of a Bayesian approach.

2.2.1 Robot kinematic modeling

Let us consider a robot with actuators non-linearities, typically transmission backlash or dead-zone. Let $q \in \mathbb{R}^n$ denote the controlled parameters, e.g., the rotation angles of the motors, and $x \in \mathbb{R}^m$ denote the robot tip pose, with $m \leqslant 6$. At discrete time k, the system model can be obtained by combining dynamic nonlinearity due to backlash, represented by a function $\mathcal{B}$, and static non-linearities resulting from the robot kinematics and denoted as $\mathcal{K}$

$$x_k = \mathcal{K}(c_k) \tag{2.1}$$
$$c_k = \mathcal{B}(\theta_k, c_{k-1}, q_k) \tag{2.2}$$

where $c_k = [c_{1,k} \ \dots \ c_{n,k}]^T \in \mathbb{R}^n$ is the output of the backlash model, and $\theta_k = [\theta_{1,k}^T \ \dots \ \theta_{n,k}^T]^T$ is the set of parameters that characterize the dynamic non-linearities in $\mathcal{B}(\cdot)$. The parameter estimates vary during their identification, at the end of which they should converge to constant values. However, for the sake of clarity in the notations, the time-dependence of model parameters is omitted in the following. It is assumed that backlash elements are independent from one another. Then, the output of the i^{th} backlash element $c_{i,k}$ is only function of the previous output $c_{i,k-1}$ and of the associated system input $q_{i,k}$, such that $c_{i,k} = \mathcal{B}_i(\theta_i, c_{i,k-1}, q_{i,k})$ as depicted in Figure 2.5.

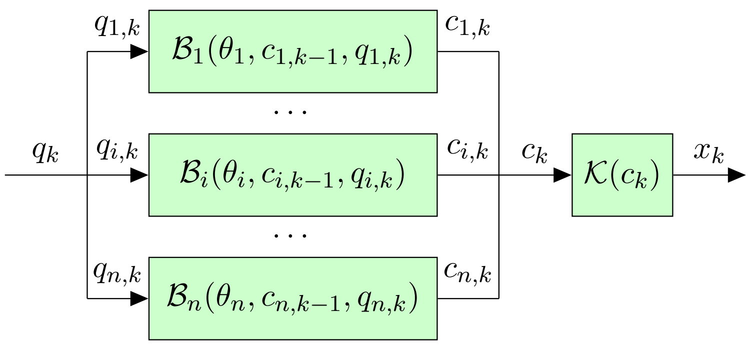


Figure 2.5: Kinematic chain of an n-DOF robot with actuation backlash.

It is important to notice that the number of parameters, and then the dimension of vector θ depend on the system non-linearities and on the model refinement. In this work, each backlash element $\mathcal{B}_i(\cdot)$ models a classical hysteresis characteristics. This corresponds to experimental measurements that were acquired with the STRAS robotized flexible instruments (Zorn et al. 2018), plotted in Figure 2.6, left. One can observe that these experimental data can be approximated by a model with linear branches, as illustrated in Figure 2.6, right.

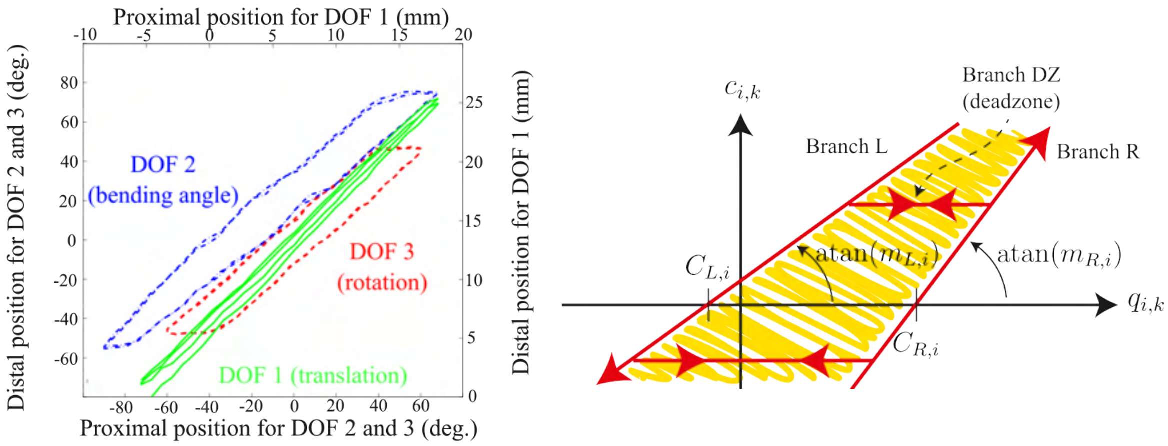


Figure 2.6: Left: real data illustrating backlash characteristics for three DOF of a robotized endoscope, acquired with the STRAS robotic endoscope (Zorn et al. 2018). Red is for rotation, blue for bending and green for translation. Right: backlash model $\mathcal{B}_i(\cdot)$ relating the i^{th} actuator input and output.

The associated model is written as

$$\mathcal{B}_i(\theta_i, c_{i,k-1}, q_{i,k}) = \begin{cases} m_{L,i}\left(q_{i,k} - C_{L,i}\right), & \text{if } q_{i,k} \leqslant \frac{c_{i,k-1}}{m_{L,i}} + C_{L,i} \qquad \text{(L)} \\ & \text{and } q_{i,k} \leqslant q_{i,k-1} \\ m_{R,i}\left(q_{i,k} - C_{R,i}\right), & \text{if } q_{i,k} \geqslant \frac{c_{i,k-1}}{m_{R,i}} + C_{R,i} \qquad \text{(R)} \\ & \text{and } q_{i,k} \geqslant q_{i,k-1} \\ c_{i,k-1}, & \text{otherwise} \qquad \text{(DZ)} \end{cases} \tag{2.3}$$

where $\theta_i = [C_{R,i}\ C_{L,i}\ m_{R,i}\ m_{L,i}]^T$ is the set of parameters characterizing the i^{th} backlash element. Note that this can be adapted if the considered backlash model is different or if additional assumptions are made, e.g., if the branches (L) and (R) are parallel. Only one branch is active at any time, depending on the direction and history of actuator motion, such that the transformation from $q_{i,k}$ to $c_{i,k}$ is the linear function associated to that branch. When the branch (DZ) is active (i.e., when crossing the dead zone), $\dot{c}_{i,k} = 0$ even if there is actuator motion.

2.2.2 State and parameters estimation

Let $\mathcal{X}_k = [\mathcal{X}_{1,k}^T \ \ldots \ \mathcal{X}_{n,k}^T]^T$ represent the state of the robot with transmission backlash, where $\mathcal{X}_{i,k} = [c_{i,k}^T \ \ \theta_i^T]^T$. From q_k and the backlash model of Equation (2.3), the state transition can be modeled as

$$\mathcal{X}_k = f(\mathcal{X}_{k-1}, q_k) + w_k \tag{2.4}$$

where $w_k \sim \mathcal{N}(0, Q_k)$ is an additive Gaussian process noise of covariance Q_k and each element of vector $f(x_{k-1}, q_k)$ is defined by

$$f_i(\mathcal{X}_{i,k-1}, q_{i,k}) = \begin{bmatrix} \mathcal{B}_i(\mathcal{X}_{i,k-1}, q_{i,k}) \\ \theta_i \end{bmatrix} \tag{2.5}$$

Similarly, an observation model relates the current state $\mathcal{X}_k$ to measurements, gathered in vector z_k such that

$$z_k = h(\mathcal{X}_k) + v_k \tag{2.6}$$

where $v_k \sim \mathcal{N}(0, R_k)$ is an additive Gaussian measurement noise of covariance R_k. Function $h(\cdot)$ may result from measurements provided by a 3D sensor measuring the tip pose, when available (e.g., a magnetic sensor). In this case, $h(\mathcal{X}_k) = x_k$ where x_k is the Cartesian position of the tip of the robot defined by Equation (2.1). Alternatively, $h(\cdot)$ can be the projection into the endoscopic camera image of fiducials attached to the robot tip. In this case, the measurement z_k corresponds to the coordinates of the segmented fiducials in the image (Reilink et al. 2013b; Poignonec et al. 2020).

Regardless of the nature of the observations and associated observation model $h(\cdot)$, measurements can be exploited to estimate $\mathcal{X}_k$, which includes the unknown model parameters θ. However, due to the switching conditions in Equation (2.3), every function $\mathcal{B}_i(\cdot)$, and by extension the transition model (see Equation 2.4), are discontinuous. Furthermore, only a subset of the θ_i parameters impacts the model at a given time. Assuming that the c_k variables are observable given the measurements z_k, the state $\mathcal{X}_k$ is still only partially observable and identifiable at any given time, since the subset of parameters appearing only in momentarily inactive branches will be unidentifiable.

A so-called Discontinuous Extended Kalman Filter (DEKF) was proposed by Chatzis et al. (2017b) to implement Bayesian filtering (i.e., EKF or UKF, see Chatzis et al. 2017b; Chatzis et al. 2017a) on switching systems that, although discontinuous, are composed of composite branches that are smooth functions of the state and input. Before each update of the estimate covariance, the DEKF decomposes the predicted state $\hat{\mathcal{X}}_{k|k-1}$ between observable and unobservable subsets, denoted as $\hat{\mathcal{X}}^o_{k|k-1}$ and $\hat{\mathcal{X}}^u_{k|k-1}$. To do so, a row ordering matrix T_k is defined (see Figure 2.7) at each time step based on which state variables are observable as proposed by Chatzis et al. (2017b) such that

$$\begin{bmatrix} \mathcal{X}^o_{k|k-1} \\ \mathcal{X}^u_{k|k-1} \end{bmatrix} = T_k \hat{\mathcal{X}}_{k|k-1} \tag{2.7}$$

The subscript conventions $\mathcal{X}_{k|k-1}$ and $\mathcal{X}_{k|k}$ denote the current prior and posterior estimates respectively.

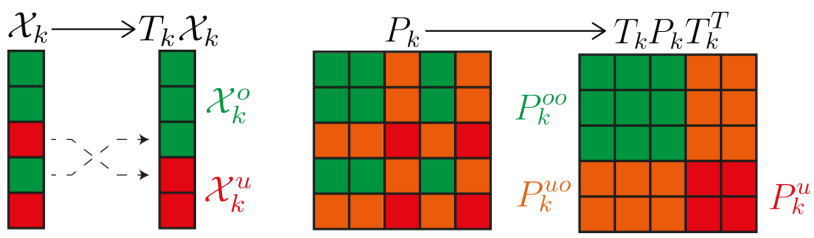


Figure 2.7: Using a row ordering matrix T as proposed in Chatzis et al. (2017b), the state $\mathcal{X}$ can be separated into observable and unobservable subsets denoted as $\mathcal{X}^o$ and $\mathcal{X}^u$, respectively. The state estimation covariance P matrix, among others, can likewise be decomposed.

The observable state variables and the associated covariance terms are updated using classical EKF equations. The unobservable state variables estimation and their covariance are kept unchanged, but the cross-covariance is updated using the update rule of the Schmit Kalman filter. Not updating the estimation of unobservable state variables prevents them from diverging, while the cross-covariance update allows the filter to retain a consistent covariance for the state estimation.

In the following, the DEKF filter is used to estimate the backlash parameters online. As the tuning of the process noise covariance matrix Q_k is difficult and very task-dependent, the DEKF is endowed with a fading memory mechanism such that older measurements are progressively forgotten (Simon 2010). Algorithm 1 details one step of the proposed fading memory DEKF (FM-DEKF) where the fading factor $\alpha \geqslant 0$ is introduced at line 10.

Algorithm 1 One step of fading memory DEKF (FM-DEKF)

Input: $\hat{\mathcal{X}}_{k-1|k-1}$, $P_{k-1|k-1}$, q_k, z_k
Output: $\hat{\mathcal{X}}_{k|k}$, $P_{k|k}$

1: **Initialisation:** $\hat{\mathcal{X}}_{0|0} \leftarrow \mathbb{E}[\mathcal{X}_0]$, $P_{0|0} \leftarrow Var[\mathcal{X}_0]$
2: **Current state prediction** :
3: $\hat{\mathcal{X}}_{k|k-1} \leftarrow f(\hat{\mathcal{X}}_{k-1|k-1}, q_k)$ ▷ Update active branch(es)
4: $F_{k-1} \leftarrow \frac{\partial f(\mathcal{X},q)}{\partial \mathcal{X}} \big|_{\mathcal{X}=\hat{\mathcal{X}}_{k-1|k-1}, q=q_k}$
5: **Decomposition observable/unobservable** :
6: Compute the row ordering matrix T_k
7: $\begin{bmatrix} \mathcal{X}^o_{k|k-1} \\ \mathcal{X}^u_{k|k-1} \end{bmatrix} \leftarrow T_k \hat{\mathcal{X}}_{k|k-1}$; $\begin{bmatrix} P^{oo}_{k-1|k-1} & (P^{uo}_{k-1|k-1})^T \\ P^{uo}_{k-1|k-1} & P^{uu}_{k-1|k-1} \end{bmatrix} \leftarrow T_k P_{k-1|k-1} T_k^T$;
8: $F^{oo}_{k-1} \leftarrow \cdots$, $Q^{oo}_k \leftarrow \cdots$
9: **Covariance propagation** :
10: $P^{oo}_{k|k-1} \leftarrow (1+\alpha) F^{oo}_{k-1} P^{oo}_{k-1|k-1} (F^{oo}_{k-1})^T + Q^{oo}_k$
11: $P^{uo}_{k|k-1} \leftarrow P^{uo}_{k-1|k-1} (F^{oo}_{k-1})^T$
12: **Measurement update** :
13: $H_k \leftarrow \frac{\partial h(\mathcal{X})}{\partial \mathcal{X}} \big|_{\mathcal{X}=\hat{\mathcal{X}}_{k|k-1}}$, $\begin{bmatrix} H^o_k \; H^u_k \end{bmatrix} \leftarrow \left(T_k H_k^T\right)^T$
14: $K_k \leftarrow P^{oo}_{k|k-1} H^o \left(H^o_k P^{oo}_{k|k-1} (H^o_k)^T + R_k\right)^{-1}$
15: $\mathcal{X}^o_{k|k} \leftarrow \mathcal{X}^o_{k|k-1} + K_k(z_k - h(\mathcal{X}^o_{k|k-1}))$ ▷ See note below
16: $P^{oo}_{k|k} \leftarrow P^{oo}_{k|k-1} - K_k H^o_k P^{oo}_{k|k-1}$
17: $P^{uo}_{k|k} \leftarrow \left((P^{uo}_{k|k})^T - K_k H^o_k (P^{uo}_{k|k})^T\right)^T$
18: $\mathcal{X}^u_{k|k} \leftarrow \mathcal{X}^u_{k|k-1}$, $P^{uu}_{k|k} \leftarrow P^{uu}_{k-1|k-1}$ ▷ No update
19: Recompose $\hat{\mathcal{X}}_{k|k}$ and $P_{k|k}$

As only the observable state variables impact the output of the (estimated) model, $h(\mathcal{X}^o_{k|k-1})$ is in practice $h(\hat{\mathcal{X}}_{k|k-1})$ where $\hat{\mathcal{X}}_{k|k-1} = T_k^T [\, (\mathcal{X}^o_{k|k-1})^T \; (\mathcal{X}^u_{k|k-1})^T \,]^T$.

2.3 One DOF illustration

The learning method is first applied to a one-dimensional case study to illustrate the different features of the algorithm. We also discuss the advantages of the proposed method over a classical EKF filter and its main limitations.

2.3.1 One-dimensional simulation

Let $c_k \in \mathbb{R}$ be the output of a backlash element as defined in Equation (2.3). The actuator input position $q_k \in \mathbb{R}$ is sampled from an arbitrary continuous sequence of values that are generated by the superposition of several periodic functions. The values of q_k are kept in the range $[-1; 1]$ (see Figure 2.8, left). In the following numerical examples, we do not mention the units of the reported values that could describe an angle, a distance, etc. The ground truth backlash element has constant parameters $C_L = -0.15$, $C_R = 0.25$, $m_L = 1.15$, and $m_R = 0.9$. Its simulated output c_k computed from q_k, Equation (2.3), for an initial value $c_0 = 0$ is reported in Figure 2.8 (left) along with the hysteretic cycle (right).

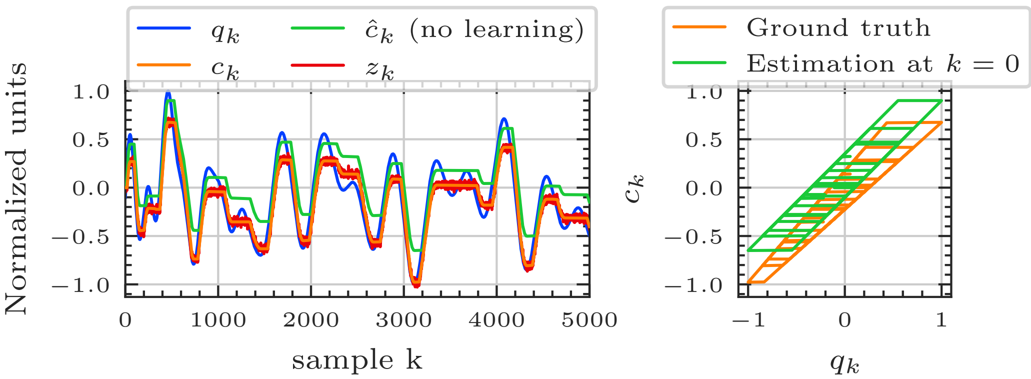


Figure 2.8: Left: input actuator position q_k, backlash element output c_k, and noisy measurement z_k used for the simulation. The estimation of c_k computed using the initial model estimation is reported. Right: input-output characteristics of the backlash element from the ground truth model and from the estimated model at $k = 0$.

The online model identification consists in estimating $\mathcal{X}_k = [c_k \; C_L \; C_R \; m_L \; m_R]^T$ given incorrect initial parameters estimates assuming that the output is measured but noisy, i.e., $z_k = h(\mathcal{X}_k) + v_k$ and $h(\mathcal{X}_k) = c_k$. The measurement noise v_k is sampled from a zero-mean Gaussian distribution of standard deviation $\sigma_v = 0.02$. The estimated state is initialized with $\hat{\mathcal{X}}_{0|0} = [0, \; -0.35, \; 0.1, \; 1, \; 1]^T$, which results in a backlash characteristics quite different from the one corresponding to the ground truth, plotted in Figure 2.8 (right). The FM-DEKF was implemented with hyper-parameters

$$\alpha = 2.10^{-3} \qquad P_{0|0} = 1.10^{-3} I_{5\times5}$$
$$R_k = (5.10^{-2})^2 \qquad Q_k = 0_{5\times5}$$

where $I_{n\times n}$ is the identity matrix of dimension n and $0_{n\times n}$ the null matrix of dimension n. The filter estimated state variable c_k is reported in Figure 2.9 along with the parameter estimation errors. Modeling errors are significantly reduced by the filter such that the mean state estimation residual error (absolute value) is $\tilde{\mathcal{X}}_k = \mathcal{X}_k - \hat{\mathcal{X}}_k = [37.10^{-5}, 26.10^{-5}, 21.10^{-5}, 15.10^{-3}, 16.10^{-3}]$ for $k \in [4000; 5000]$ in normalized units. The active branch of the backlash element is also reported in Figure 2.9. As desired, any unobservable state variable remains constant until it becomes observable again (e.g., C_L and m_L are only observable when branch (L) is active). Model parameter errors are reduced by 99.3% (i.e. error roughly divided by 100) on average by the end of the identification process (see Figure 2.9, center). When using the initial model estimation only (see Figure 2.8 for initial model prediction), the mean and standard deviation of the c_k prediction errors over the whole simulation is 0.19 ± 0.05 in normalized units. This error is reduced here down to 0.0037 ± 0.009 when using the FM-DEKF.

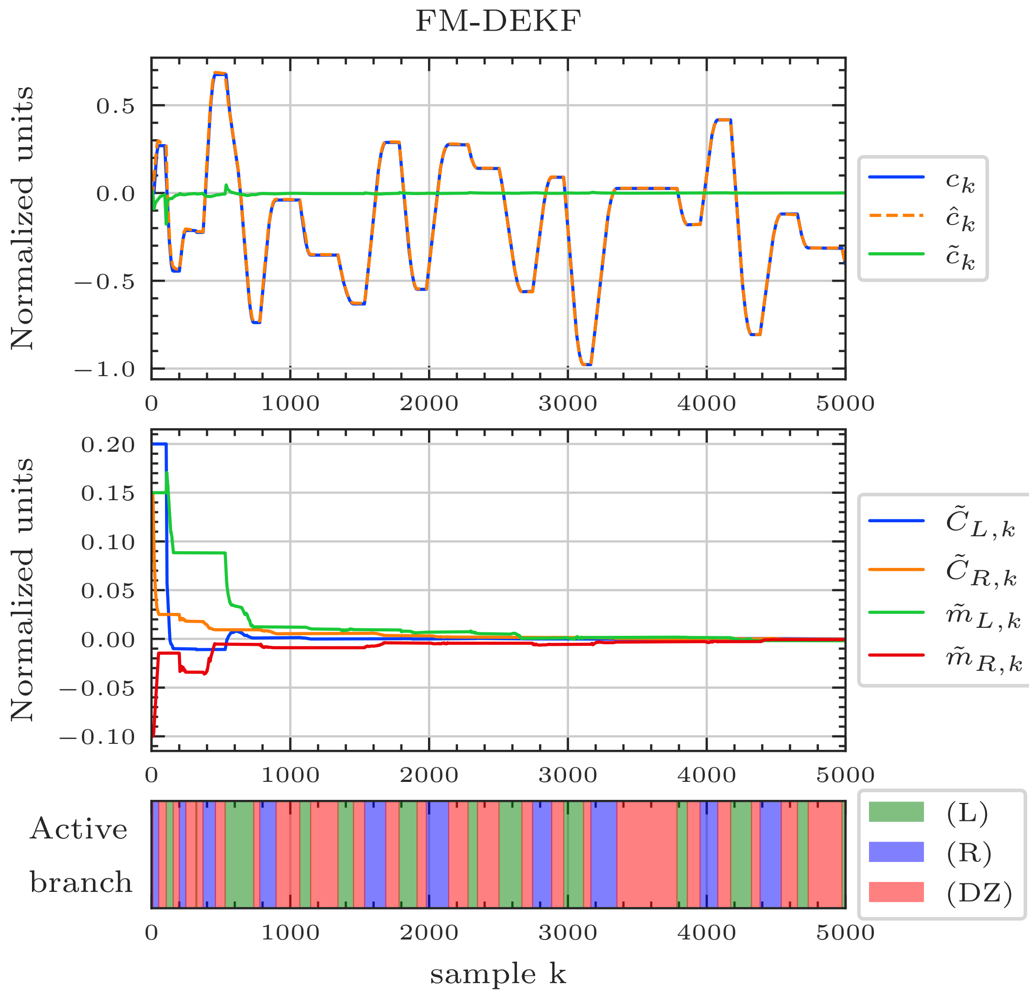


Figure 2.9: Top: State variable c_k, its estimation $\hat{c}_{k|k}$ by the FM-DEKF, and estimation errors $\tilde{c}_k = c_k - \hat{c}_{k|k}$. Center: Estimation errors of the state variables associated with the model parameters. Bottom: estimated active branch of the backlash element.

2.3.2 Comparison with classical EKF

For comparison purposes, a classical fading memory EKF (FM-EKF) was also implemented. Its design is identical to the FM-DEKF, but without the observability management (i.e., supposing $\mathcal{X}_k^o = \mathcal{X}_k$). The performance of such a FM-EKF was evaluated with the same hyperparameters α, $P_{0|0}$, $\hat{\mathcal{X}}_{0|0}$, and Q_k, and in the same scenario. Figure 2.10 (top) reports the state variables estimation errors associated with the backlash model. The errors converge towards 0 even faster, if less smoothly, than with the FM-DEKF (see Figure 2.9, center).

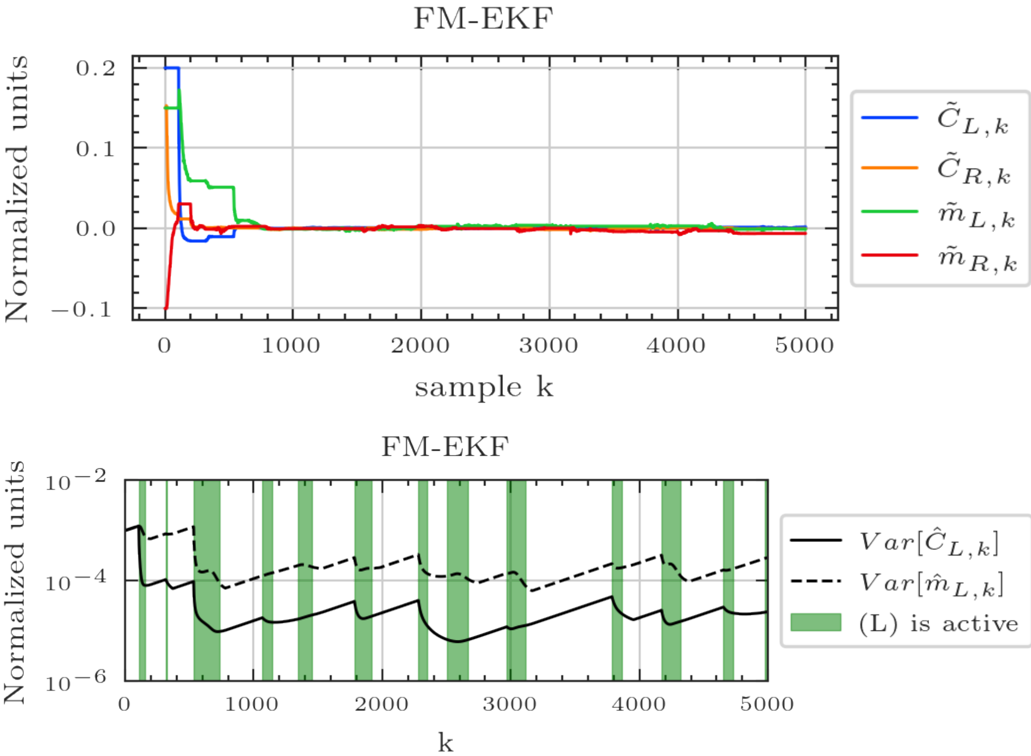


Figure 2.10: Left: Estimation errors of the state variables associated with the model parameters when using a classical FM-EKF. Right: The variances of the estimates are displayed on a logarithmic scale for the two parameters associated with the branch (L).

However, as the parameters observability is not taken into account, the covariance of an estimate grows whenever the variable is not observable (see Figure 2.10, bottom). This is to be expected since process noise is continually injected through the fading memory mechanism regardless of whether observations yield information. As each branch is persistently activated (i.e., there is a constant motion on each DOF and its direction varies), such periods of undesirable estimation covariance increase are limited in duration. This can nonetheless lead to inaccurate measurement updates and even to the divergence of the filter if the covariance increases in such a way for too long.

Although the FM-DEKF does not perform better than the FM-EKF in terms of convergence rate, it increases the robustness of the overall learning process. A thorough sensibility and observability analysis are out of scope, but it should be noted that the filter convergence is contingent on persistent actuator excitation (i.e., there must be actuator displacements).

2.4 Simulation results

In order to demonstrate the efficiency of the FM-DEKF method to learn the input backlash model of a cable-driven robot, the filter performance is evaluated for a simulated 3 DOF robotized endoscope, whose kinematics is described Figure 2.11, corresponding to a common architecture for robotized endoscopes (Zorn et al. 2018). In this case, the $c_k \in \mathbb{R}^3$ variables are the axial translation, the bending of a flexible section, and the rotation around the Z axis.

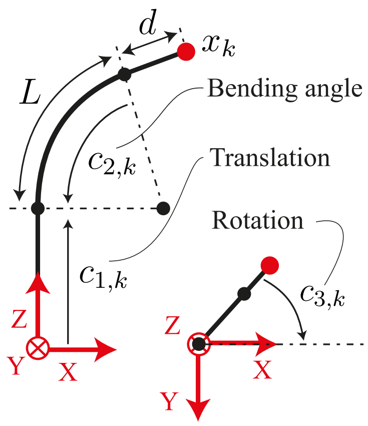


Figure 2.11: Endoscopic robot with 3 DOF, characterized by its axial translation along Z, the bending angle of the bending section of the body, and the rotation w.r.t Z axis.

The three backlash elements associated with the three actuators of the robot are modeled using Equation (2.3) such that 12 initially incorrect model parameters have to be estimated. The online model identification procedure is performed with the presented FM-DEKF approach from a noisy measure of the distal Cartesian position of the robot $x_k \in \mathbb{R}^3$. Although model parameters used for the simulation are realistic, we do not claim that they fully capture the complexity of real-world cable-actuated endoscopic robots as will be discussed at the end of Part I. This is foremost a proof of concept that demonstrates the advantages of the proposed method.

2.4.1 Simulation setup

The bending angle $c_{2,k}$ is actuated using two antagonist bowden cables such that the actuator position $q_{2,k}$ is a cable length variation. As depicted in Figure 2.6 (left), backlash phenomena can appear in all three DOF of the robot. The kinematic chain depicted in Figure 2.11 can be modeled using a constant curvature model of the bending section (Webster III et al. 2010). Under this approximation, there is an analytical form for the kinematic function $x_k = \mathcal{K}(c_k)$ as detailed in (Nageotte et al. 2020) and reported in the appendix (see Equation B.1).

Persistent excitation similar to the one used in Section 2.3 are generated for each actuator positions $q_{i,k}$, but adapted so that their respective ranges match that of existing robotized endoscopic tools (Zorn et al. 2018). The values of $c_{2,k}$ and $c_{3,k}$ are kept in the range $[-\pi/2; \pi/2]$ rad (see Figure 2.13, top). The observations $z_k = x_k + v_k$ are simulated by modeling v_k as zero-mean Gaussian noise $v_k \sim \mathcal{N}(0, \sigma_v^2)$ of standard deviation $\sigma_v = 0.5$ mm. Actuator positions q_k and observations z_k are sampled at 50 Hz over a time period of 180 s. The FM-DEKF is implemented with hyperparameters $\alpha = 2.10^{-3}$ (equivalent to a decay time-constant of 20 s), $R_k = (2.10^{-3})^2 I_{3\times3}$ and $Q_k = 0_{15\times15}$. The initial state estimate $\hat{\mathcal{X}}_{0|0}$ is significantly inaccurate, with estimation errors on all model parameters (see resulting characteristics in Figure 2.12). The initial state estimate covariance $P_{0|0}$ is arbitrarily computed by scaling a matrix $(2.10^{-2})^2 I_{15\times15}$ according to the respective $q_{i,k}$ and $c_{i,k}$ working ranges such that

$$\mathrm{diag}(P_{0|0}) = \begin{bmatrix}\mathrm{diag}(P_{1,0|0}) & \mathrm{diag}(P_{2,0|0}) & \mathrm{diag}(P_{3,0|0})\end{bmatrix}$$

where

$$\sqrt{\mathrm{diag}(P_{i,0|0})} = 2.10^{-2} \begin{bmatrix}\Delta c_i & \Delta q_i & \Delta q_i & \dfrac{\Delta c_i}{\Delta q_i} & \dfrac{\Delta c_i}{\Delta q_i}\end{bmatrix} \tag{2.8}$$

is a row vector containing the initial parameter standard deviations of $\hat{c}_{i,0}$, $\hat{C}_{L,i,0}$, $\hat{C}_{R,i,0}$, $\hat{m}_{L,i,0}$, and $\hat{m}_{R,i,0}$ (in this order). The function $\mathrm{diag}(\cdot)$ returns the row vector containing the diagonal terms of a matrix and the square root in Equation (2.8) is computed element-wise. Finally, Δc_i and Δq_i are the i^{th} actuator and configuration variables expected ranges, respectively.

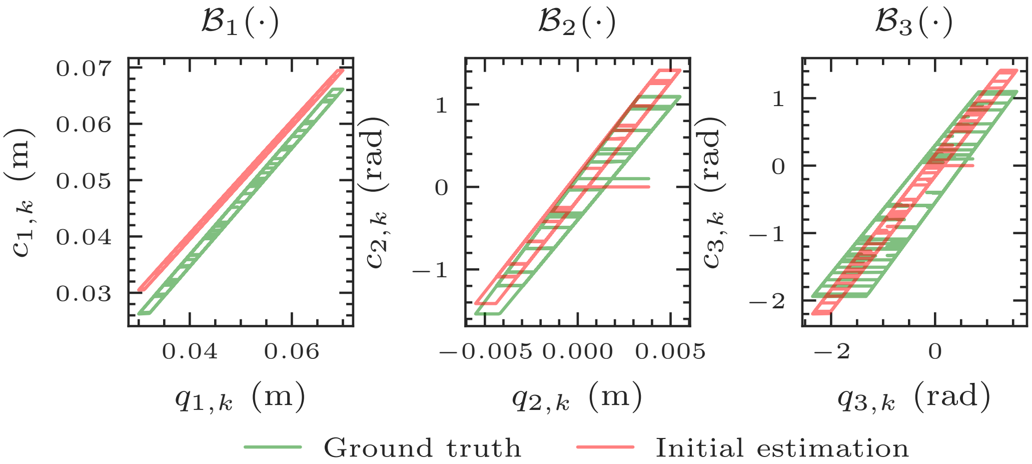


Figure 2.12: Input-output characteristics of the backlash elements $\mathcal{B}_i(\cdot)$ from the ground truth models and from the estimated models at $k = 0$.

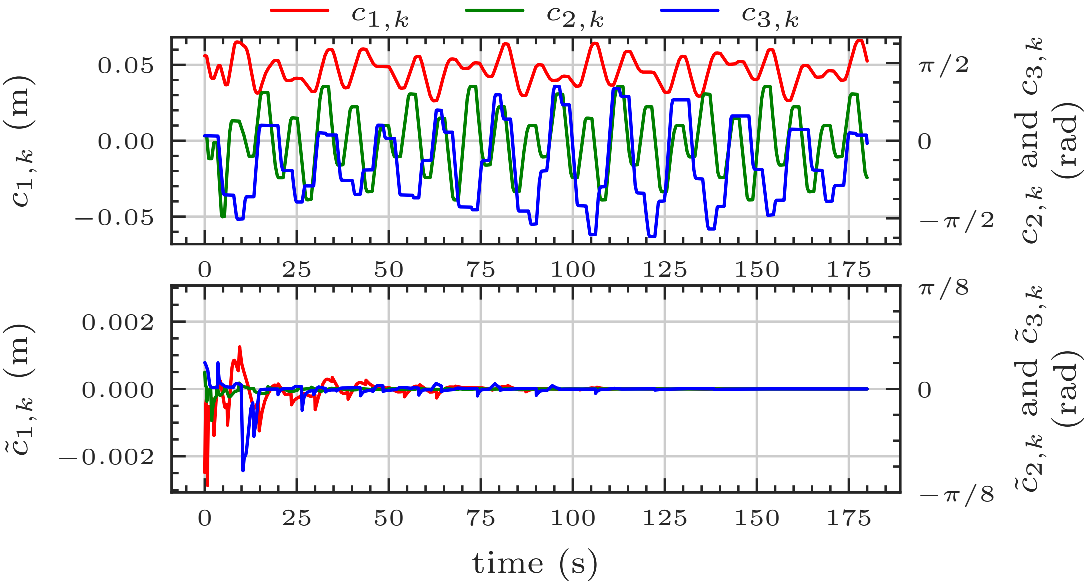


Figure 2.13: Top: Ground truth values for $c_{i,k}$. Bottom: FM-DEKF prediction error $\tilde{c}_{i,k} = c_{i,k} - \hat{c}_{i,k}$.

2.4.2 Results

All backlash model parameter errors are reduced by 80% or more after 90s of simulation (see Figure 2.14), for an average error reduction of 99.6% at the end of the simulation. Likewise, the estimated variables c_k are correctly predicted once the FM-DEKF converges (see Figure 2.13, bottom). The mean estimation error for c_k over the time period $t \in [120\text{ s}, 180\text{ s}]$ is $\begin{bmatrix}3.10^{-6}\text{ m}, & 9.10^{-5}\text{ rad}, & 6.10^{-4}\text{ rad},\end{bmatrix}$ and the same error is $\begin{bmatrix}1.10^{-4}, & 2.10^{-3}, & 8.10^{-3}\end{bmatrix}$ over the whole simulation. To put this in perspective, the mean c_k estimation error over the whole simulation would be $\begin{bmatrix}0.0038, & 0.17, & 0.23\end{bmatrix}$ if the initial model estimation was used to predict c_k in the same conditions, but without any model parameters updates.

In Figure 2.14, sudden relative error reduction can be observed at start-up for some model parameters (e.g., $\hat{C}_{L,2,k}$ or $\hat{C}_{L,3,k}$). This is partly explained by the smaller initial absolute estimation error on these parameters (see (L) branches in Figure 2.12 for $\mathcal{B}_2$ and $\mathcal{B}_3$), but also by an aggressive hyperparameter tuning. Smoother, more gradual parameters updates can be obtained by reducing the initial state covariance and choosing a smaller value for α, but this would also increase the time required for convergence.

2.5 Conclusions

We proposed to use DEKF filtering for endoscopic robot backlash model identification, allowing for the identification of more complex backlash characteristics than existing online methods that deal with this problem. The DEKF was adapted to the case of endoscopic robot model identification and a forgetting factor added to reduce the tuning

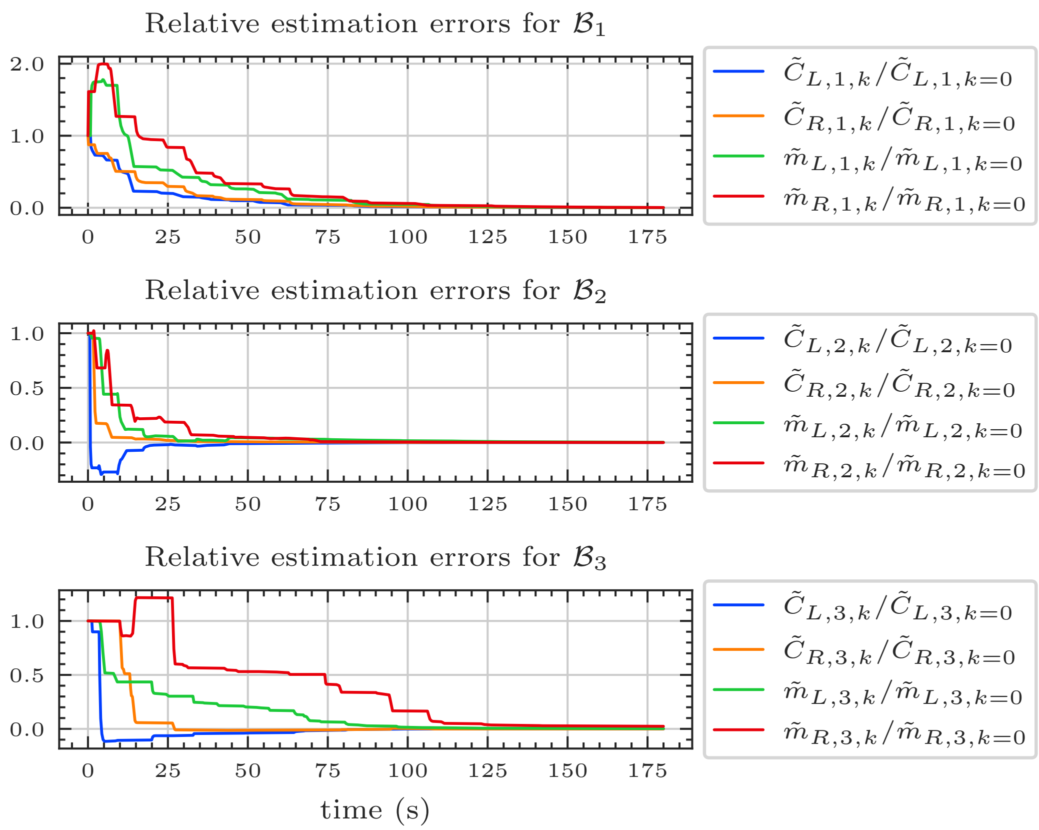


Figure 2.14: Normalized estimation error for all model parameters computed as the element-wise ratio $\tilde{\mathcal{X}}_k/\tilde{\mathcal{X}}_0$ where $\tilde{\mathcal{X}}_k = \mathcal{X}_k - \hat{\mathcal{X}}_k$.

difficulty and task dependency. A simulation of an existing endoscope was used to assess this approach on the intended use-case. Overall, the performance of the filter is adequate: provided sufficient actuators excitation and reasonable hyperparameters, the model parameters are correctly learned online in a limited time duration. The learned models are sufficient to model non-constant backlash widths found in many cable-actuated robots, but could be simplified if needed.

The main limitation of the method is the actuator excitation necessary to the filter convergence, except when only $C_{L,i}$ and $C_{R,i}$ are learned, in which case the parameters are observable regardless of the excitation. This is not an issue as long as the actuator trajectories are automatically planned (e.g., automatic calibration procedure). However, it could become a problem during the robot teleoperation by an operator. To be robust against arbitrary trajectories, the FM-DEKF should be improved to cope with adversary conditions.

In the simulation presented in Section 2.4, a measurement of the Cartesian position x_k was supposed readily available, which is unlikely in an *in-situ* medical scenario with an endoscopic robot. Measurements can sometimes be acquired from images captured by an endoscopic monocular camera in an eye-to-hand configuratio. In that case, z_k can be defined as the image measurements of several markers attached to the robot (Reilink et al. 2013b; Cabras et al. 2017) or even the result of an intermediary markerless pose estimation (Baek et al. 2020; Sestini et al. 2021). However, some endoscopic robots have an eye-in-hand camera configuration, such that the camera is at the tip of the endoscope and not directed at it, which makes direct robot pose measurements impossible. In such cases, the FM-DEKF approach can be coupled with SLAM to use endoscopic images of the environment to correct the models and the map of said environment (i.e., depth estimation) simultaneously. In the following chapter, this concept is explored to implement online *in-situ* backlash width identification on eye-in-hand endoscopic robots.

Chapter 3

Eye-in-hand endoscopic robot in-situ backlash identification

Contents

3.1 Preliminary developments for endoscope and backlash modeling 38
3.1.1 Forward kinematics of eye-in-hand endoscopes 38
3.1.2 TSM backlash modeling 38
3.1.3 Backlash model parameterization 39
3.2 Proposed backlash width identification method 40
3.2.1 Problem formulation 40
3.2.2 Comparison with visual SLAM 42
3.2.3 Choice of parameters to be learned 42
3.2.4 Practical filter implementation 43
3.3 Simulation results 44
3.3.1 Simulation setup 44
3.3.2 Backlash width estimation when $\hat{m}_{O,i}$ is correct 45
3.3.3 Robustness to incorrect $\hat{m}_{O,i}$ values 46
3.3.4 Discussion 46
3.4 Experimentation on a robotic endoscopic platform 48
3.4.1 Experimental setup and ground truth estimation 48
3.4.2 Results 50
3.4.3 Discussion 50
3.5 Conclusions 52

Existing methods for *in-situ* automatic backlash identification from endoscopic images were presented in Chapter 2, but they require a direct image measurement of the robot to estimate its pose. Thus, they are not directly applicable to eye-in-hand endoscopic robots such as conventional gastroscopes or colonoscopes (see Figure 3.1). It should be noted that backlash estimation methods based on robot pose estimation can be used for the offline identification of eye-in-hand endoscopes. For instance, a stereoscopic camera was used to estimate the pose of the endoscope and learn the inverse kinematics from data, backlash included (Aleluia Porto 2021). All the previously introduced works (see Section 2.1) could similarly be extended to the case of eye-in-hand endoscopes provided adequate external pose measurement. However, such external pose measurement will not be available *in-situ*.

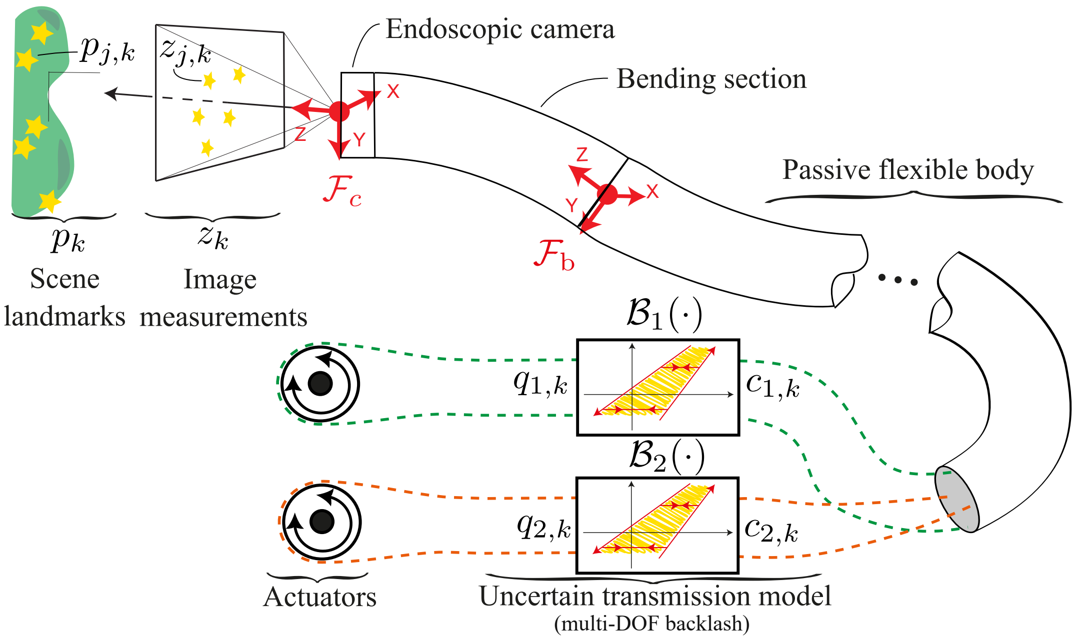


Figure 3.1: Considered scenario with an eye-in-hand endoscope. We want to automatically reconstruct the transmission's backlash model from actuator and image measurements only.

Ott et al. (2011) proposed an approach adapted to this scenario that relies on the observation of a fixed point of environment tracked in the endoscopic image. The authors measured the backlash width of the hysteretic cycle generated when tracking a feature in the endoscopic image during actuator excitation. The main limitation of this approach is that it can only be used during specific DOF-by-DOF actuator displacement sequences and, therefore, not while the robot is executing arbitrary operational motions. To the best of our knowledge, this is the only approach adapted to the *in-situ* identification of eye-in-hand endoscopes.

Therefore, in this chapter, we investigate how the backlash model can be learned *in-situ* and online from indirect measurements only, from several scene landmarks tracked in the endoscopic image. We propose to estimate simultaneously the parameters of the backlash model and the 3D position of the landmarks, provided some reasonable assumptions. In order to make the problem solvable, the backlash width model is separated from the full backlash model so that only the part strictly necessary for backlash compensation is learned. The approach is then implemented using the FM-DEKF algorithm presented in Chapter 2.

3.1 Preliminary developments for endoscope and backlash modeling

3.1.1 Forward kinematics of eye-in-hand endoscopes

The endoscope kinematics is modeled using an analytical model assuming constant curvature of the 2 DOF endoscope during the bending in two orthogonal planes (Ott et al. 2011). The actuation variables $q_k = [q_{1,k},\, q_{2,k}]^T$ are defined as proximal side cable length variations, typically driven by a pulley system. Proximal cable displacements q_k results in a distal cable displacement $c_k = [c_{1,k},\, c_{2,k}]^T$ as shown in Figure 3.1. Note that q_k and c_k would be identical in the absence of friction, cable slack, and other phenomena that tend in practice to create hysteretic behaviors. The distal side cable displacements c_k are mapped into a configuration space composed of ϕ_k, the orientation of the bending plane, and β_k, the bending angle in this plane (see Figure 3.2) such that

$$\begin{bmatrix} \phi_k \\ \beta_k \end{bmatrix} = \mathcal{K}_c(c_k) \tag{3.1}$$

as detailed in Appendix B.2. The endoscopic camera is located at the tip of the endoscope. Its Cartesian pose w.r.t. the base of the bending section, is modeled using the homogeneous transformation matrix

$${}^bT_c(c_k) = \begin{bmatrix} {}^bR_c(c_k) & {}^bt_c(c_k) \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} = \mathcal{K}_p(\phi_k, \beta_k) \tag{3.2}$$

where ${}^bt_c(c_k) \in \mathbb{R}^3$ is the Cartesian position of the camera in the frame $\mathcal{F}_b$ (see Figure 3.2, right) and ${}^bR_c(c_k) \in SO(3)$ is the orientation matrix of the camera frame $\mathcal{F}_c$ w.r.t. $\mathcal{F}_b$. The full analytical expressions of ${}^bt_c(c_k)$ and ${}^bR_c(c_k)$ are reported in Appendix B.2.

3.1.2 TSM backlash modeling

Backlash in the transmission from proximal to distal cables displacement is modeled as two independent backlash elements $\mathcal{B}_1$ and $\mathcal{B}_2$ as illustrated in figures 3.1 and 3.3. Each element (i.e., one per DOF) is described by two linear functions as was introduced in Chapter 2, hence requiring 4 model parameters per backlash element denoted $\theta_i \in \mathbb{R}^4$ (see Figure 2.6, right). This type of backlash model with two linear branches is a reasonable local approximation for real systems (Ott et al. 2011). The backlash model identification then consists in estimating the parameters $m_{L,i}$, $C_{L,i}$, $m_{R,i}$, and $C_{R,i}$ that were introduced in Equation (2.3).

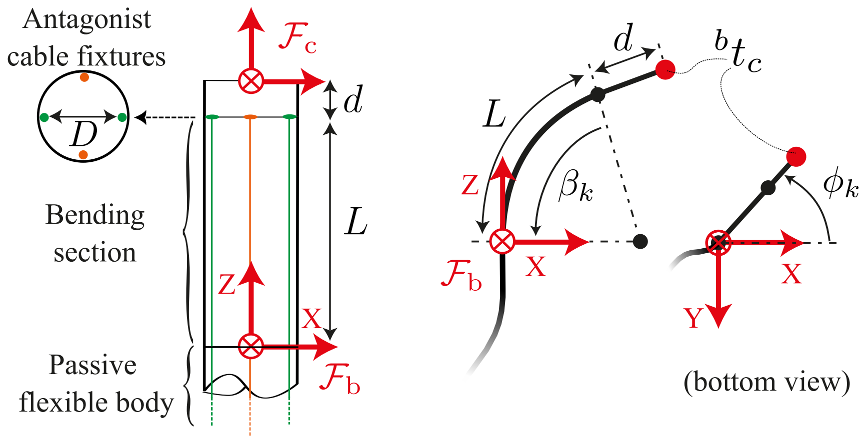


Figure 3.2: Left: TSM-actuated endoscopic robot with 2 DOF and an eye-in-hand camera configuration. The camera is located at the origin of the frame $\mathcal{F}_c$. Right: constant curvature model (Ott et al. 2011).

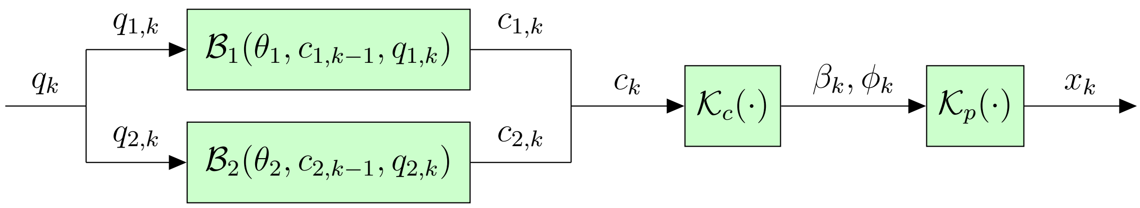


Figure 3.3: Kinematic chain of the robot with actuation backlash.

3.1.3 Backlash model parameterization

Backlash elements $\mathcal{B}_i(\cdot)$ can be decomposed between the backlash width itself (dependent on actuator position), denoted b_i, and a linear mapping from actuator space to the distal cable position space. The backlash width is most naturally expressed as a function of c_k such that

$$b_i(c_{i,k}) = \Gamma_i^A + \gamma_i^A c_{i,k} \tag{3.3}$$

where Γ_i^A and γ_i^A are combinations of the original model parameters. In order to isolate these two parameters, the backlash model (see Equation 2.3) parameterization can be modified such that

$$m_{O,i} = \frac{2m_{R,i}m_{L,i}}{m_{L,i} + m_{R,i}} \qquad \gamma_i^A = \frac{m_{L,i} - m_{R,i}}{m_{R,i}m_{L,i}}$$
$$C_{O,i} = \frac{C_{R,i} + C_{L,i}}{2} \qquad \Gamma_i^A = C_{R,i} - C_{L,i}$$

This backlash model parameterization will hereinafter be referred to as representation A. The backlash width model b_i can also be expressed as a function of the actuator position, which results in the representation B depicted in Figure 3.4:

$$\bar{q}_{i,k} = c_{i,k}m_{O,i} + C_{O,i} \tag{3.4}$$
$$b_i(\bar{q}_{i,k}) = \Gamma_i^B + \gamma_i^B \bar{q}_{i,k} \tag{3.5}$$

where $\bar{q}_{i,k}$ is the mean actuator position among those that could result in a given output $c_{i,k}$. Both representations are *in fine* equivalent: going from representation A to representation B can be seen as a projection of the backlash width into the actuator space (see Figure 3.4). The backlash width identification therefore consists in estimating Γ_i^A and γ_i^A (or Γ_i^B and γ_i^B) from available *in-situ* measurements. It should be noted that with this problem formulation we do not learn the full backlash models (i.e., 4 parameters per DOF), but only a set of parameters sufficient to describe the variable backlash width b_i.

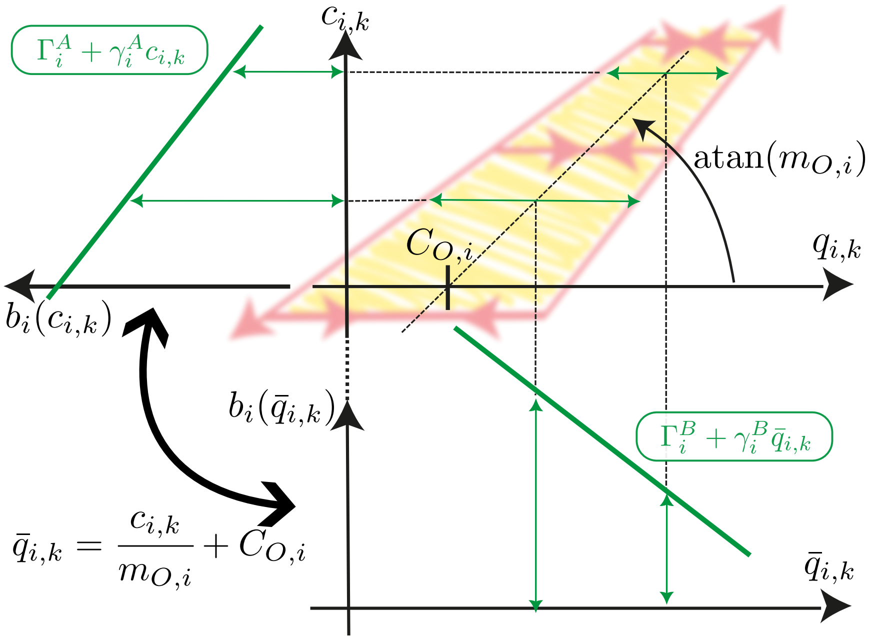


Figure 3.4: Two equivalent backlash model parameterizations where the variable backlash width (in green) is independent from the overall backlash characteristic (in red).

3.2 Proposed backlash width identification method

3.2.1 Problem formulation

If only the endoscopic images are available for learning, as is typically the case for *in-situ* estimation, then the only information comes from image measurements of the environment as depicted in Figure 3.1. Let $p_k = \left[p_{1,k}^T \cdots p_{j,k}^T \cdots \; p_{N,k}^T\right]^T$, with $p_{j,k} \in \mathbb{R}^3$, be the Cartesian positions of N scene landmarks tracked in the endoscopic camera image such that their measured positions in the image are z_k (see Figure 3.1). From a sequence of measurements z_k, we wish to reconstruct the (possibly variable) backlash width of the robot under the following assumptions:

- the intrinsic model of the camera has been correctly identified during an offline calibration;

- the geometric parameters of the distal bending section are known (lengths L and d, see Figure 3.2);
- the environment is rigid such that the distances between landmarks $p_{j,k}$ remain constant. Additionally, the endoscope body is not displaced during the identification procedure (i.e., $p_{j,k}$ constant in $\mathcal{F}_b$).

Let $p_{k/\mathcal{F}_b}$ and $p_{k/\mathcal{F}_c}$ denote the landmarks positions expressed in the base (i.e., proximal) frame $\mathcal{F}_b$ of the endoscope and distal camera frame $\mathcal{F}_c$ respectively (see Figure 3.1). The measurements z_k can be modeled using a pinhole projection model and the robot distal kinematics such that

$$z_k = \begin{bmatrix} z_{1,k}^T & \cdots & z_{j,k}^T & \cdots & z_{N,k}^T \end{bmatrix}^T \tag{3.6}$$

$$z_{j,k} = \begin{bmatrix} u_0 + K_x \dfrac{p^x_{j,k/\mathcal{F}_c}}{p^z_{j,k/\mathcal{F}_c}} \\ v_0 + K_y \dfrac{p^y_{j,k/\mathcal{F}_c}}{p^z_{j,k/\mathcal{F}_c}} \end{bmatrix} \tag{3.7}$$

where u_0, v_0, K_x, and K_y are camera intrinsic model parameters (central point coordinates and magnification factors, respectively) and

$$p_{j,k/\mathcal{F}_c} = \begin{bmatrix} p^x_{j,k/\mathcal{F}_c} \\ p^y_{j,k/\mathcal{F}_c} \\ p^z_{j,k/\mathcal{F}_c} \end{bmatrix} = \left({}^bR_c(c_k)\right)^T \left(p_{,k/\mathcal{F}_b} - {}^bt_c(c_k)\right) \tag{3.8}$$

Then, we define the state $\mathcal{X}_k$ as the concatenation of the distal cable displacement c_k, the parameters $\theta_k = [\theta_{1,k}^T\,\theta_{2,k}^T]^T$, and the landmark Cartesian positions expressed in $\mathcal{F}_b$ such that

$$\mathcal{X}_k = \begin{bmatrix} c_k^T & \theta_k^T & p_{k/\mathcal{F}_b}^T \end{bmatrix}^T = f(\mathcal{X}_{k-1}, q_k) + w_k,\ \ w_k \sim \mathcal{N}(0, Q_k) \tag{3.9}$$

and

$$z_k = h(\mathcal{X}_k) + v_k,\ \ v_k \sim \mathcal{N}(0, R_k) \tag{3.10}$$

where $h(\mathcal{X}_k)$ is derived from equations (3.7) and (3.8). Note that $\theta_{1,k}$ and $\theta_{2,k}$ are the parameters of $\mathcal{B}_1$ and $\mathcal{B}_2$, respectively. The matrix Q_k encodes the covariance of the process noise w_k and the matrix R_k that of the measurement noise v_k. The state transition model is defined as

$$f(\mathcal{X}_{k-1}, q_k) = \begin{bmatrix} \mathcal{B}_1(\theta_{1,k}, c_{1,k-1}, q_{1,k}) \\ \mathcal{B}_2(\theta_{2,k}, c_{2,k-1}, q_{2,k}) \\ \theta_k \\ p_{k/\mathcal{F}_b} \end{bmatrix} \tag{3.11}$$

Provided system identifiability and sufficient actuators excitation, Bayesian filtering approaches can then be used to learn online the parameters θ_k considering the state transition and the observation models defined by equations (3.9) and (3.10), respectively. We propose to use an EKF to learn the backlash width parameters. Nevertheless, a standard EKF cannot be used due to the discontinuous nature of the backlash model and its parameters local unidentifiability. Therefore, a practical implementation that can cope with the presence of backlash is detailed in Section 3.2.4.

3.2.2 Comparison with visual SLAM

The proposed problem formulation is similar to (and inspired by) SLAM. Monocular SLAM has been applied to medical endoscopic images (Grasa et al. 2014; Mahmoud et al. 2019) to reconstruct the 3D shape of the environment and the pose of the camera during motion, independently from the kinematic model of the robot (i.e., backlash and robot kinematics). However, reconstructing the camera trajectory in such a way is not directly useful to learn the backlash in the transmission, as it is the position of the camera w.r.t. the base of the kinematic chain that would be necessary, an information not recovered by the SLAM. Nonetheless, the backlash identification approach proposed in this paper can be seen as a SLAM-like problem, but where the estimated state contains the parameters of the TSM backlash models and not the pose of the camera w.r.t. the environment.

3.2.3 Choice of parameters to be learned

Intuitively, the configuration of the endoscope is hard to retrieve from image measurements since, for small displacements, the camera moves on a sphere (see Figure 3.5 for planar illustration). As a result, the absolute position on this sphere (parameterized by $C_{R,i}$ and $C_{L,i}$ or by $C_{O,i}$) cannot be estimated because the configuration does not impact the apparent motion of scene landmarks (see Figure 3.5, right). To a lesser degree, it is also challenging to retrieve the slope parameters (i.e., $m_{R,i}$ and $m_{L,i}$ or $m_{O,i}$) due to model sensitivity issues. Locally, a variation of the landmarks depth and a variation of the slope parameter(s) have the same effect on image measurements (see Figure 3.5, right).

Ideally, complete and correct model parameters identification would then require large endoscope displacements (i.e., large bending amplitude), which is not realistic in medical applications. On the contrary, small endoscope motions should be preferred so as not to deform the environment. With only small endoscope displacements, the estimation of the complete model would therefore be highly sensitive to measurement noise, unmodeled endoscope motions, and modeling errors in general.

Therefore, only the backlash widths b_i will be identified using the alternative parameterization introduced in Section 3.1.2, leading to the set of estimated parameters $\theta_k = [\gamma^A_{1,k}\, \Gamma^A_{1,k}\, \gamma^A_{2,k}\, \Gamma^A_{2,k}]^T$. We choose to use representation A because it is more intuitive and results in a more compact representation. Nevertheless, even if representation B is more cumbersome, it is relevant to practical applications such as backlash compensation that requires backlash models expressed in the actuator space. We will therefore implement the identification method using representation A, but subsequently express the results using representation B. The choice is motivated by the fact that the values of $\gamma^A_{i,k}$ and $\Gamma^A_{i,k}$ are dependent on the estimate of $c_{i,k}$, which will most likely be inaccurate since the parameters $m_{O,i}$ and $C_{O,i}$ are not identified, hence not corrected.

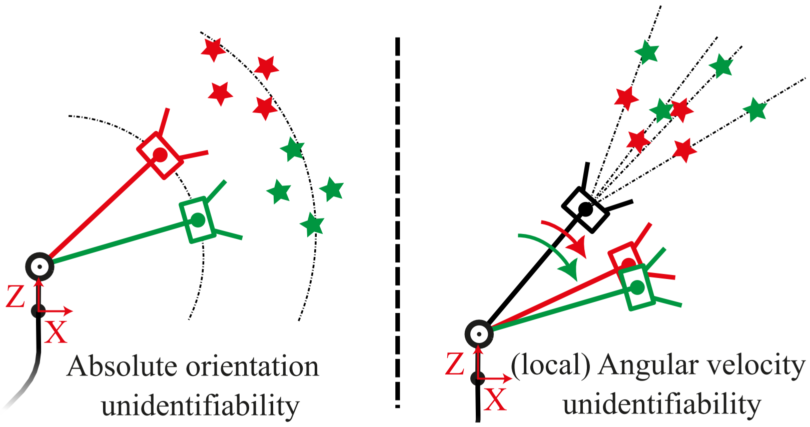


Figure 3.5: Intuitions behind identification issues for the absolute orientation and the angular velocity. They are due to the fact that for small displacements (i.e., small bending angle variations), the endoscopic robot can be approximated by a simple revolute joint in 2D. The illustration shows states that result in the same image measurements during the EKF-SLAM identification procedure.

3.2.4 Practical filter implementation

We use the fading memory discontinuous EKF filter introduced in Chapter 2 to solve the identification problem considering the state

$$\mathcal{X}_k = \begin{bmatrix} c_k^T & \theta_k^T & p_{k/\mathcal{F}_b}^T \end{bmatrix}^T \in \mathbb{R}^9 \tag{3.12}$$

where

$$\begin{aligned} c_k &= \begin{bmatrix} c_{1,k} & c_{2,k} \end{bmatrix}^T \\ \theta_k &= \begin{bmatrix} \gamma_{1,k}^A & \Gamma_{1,k}^A & \gamma_{2,k}^A & \Gamma_{2,k}^A \end{bmatrix}^T \\ p_{k/\mathcal{F}_b} &= \begin{bmatrix} p_{1,k/\mathcal{F}_b}^x & p_{1,k/\mathcal{F}_b}^y & p_{1,k/\mathcal{F}_b}^z \end{bmatrix}^T \end{aligned} \tag{3.13}$$

such that only a single landmark is tracked in the image, which is in practice sufficient. Given the first image measurements z_0 and the initial estimation of c_0, the landmark position estimate is initialized as

$$\hat{p}_{1,0} = g(z_{1,0}, \hat{c}_0, \hat{p}_{1,0/\mathcal{F}_c}^z) \tag{3.14}$$

where $\hat{p}_{1,0/\mathcal{F}_c}^z$ is an arbitrary initial depth estimate and the function $g(\cdot)$ is obtained by inversion of equations (3.7) and (3.8) such that

$$g(z_{1,0}, \hat{c}_0, \hat{p}_{1,0/\mathcal{F}_c}^z) = {}^bR_c(\hat{c}_0) \begin{bmatrix} \hat{p}_{1,0/\mathcal{F}_c}^z \begin{bmatrix} 1/K_x & 0 \\ 0 & 1/K_y \end{bmatrix} \left(z_{1,0} - \begin{bmatrix} u_0 \\ v_o \end{bmatrix} \right) \\ \hat{p}_{1,0/\mathcal{F}_c}^z \end{bmatrix} + {}^bt_c(\hat{c}_0) \tag{3.15}$$

Finally, the covariance of the initial state estimation is computed as

$$P_{0|0} = \begin{bmatrix} \mathrm{Var}(\hat{c}_0) & & 0 \\ & \mathrm{Var}(\hat{\theta}_0) & \\ 0 & & P_{1,0|0} \end{bmatrix} \tag{3.16}$$

where $\mathrm{Var}(\cdot)$ denotes the variance-covariance matrix and $P_{1,0|0}$ is the covariance of the landmark initial estimate, computed by propagation of the initial robot configuration uncertainty and measurements noise covariance such that

$$P_{1,0|0} = G_0 \begin{bmatrix} R_0 & 0 & 0 \\ 0 & \mathrm{Var}(\hat{c}_0) & 0 \\ 0 & 0 & \mathrm{Var}(\hat{p}^z_{1,0/\mathcal{F}_c}) \end{bmatrix} G_0^T \tag{3.17}$$

with

$$G_0 = \begin{bmatrix} \frac{\partial g(\cdot)}{\partial z_{1,0}} & \frac{\partial g(\cdot)}{\partial \hat{c}_0} & \frac{\partial g(\cdot)}{\partial \hat{p}^z_{1,0/\mathcal{F}_c}} \end{bmatrix} \tag{3.18}$$

3.3 Simulation results

The method is first illustrated in a simulated, but realistic scenario. This simulation is well suited to validate the method and allows us to discuss the evolution of internal variables not typically available during an experiment on a real robot as, for instance, the ground truth distal cable displacements.

3.3.1 Simulation setup

An environment as illustrated in Figure 3.1 is simulated. It contains one landmark positioned at $p_{1,k/\mathcal{F}_b} = [10, 10, 250]$ mm – i.e., 150 mm in front of the endoscopic camera when in the straight position. The dimensions of the simulated endoscope are $D = 12$ mm, $d = 20$ mm, and $L = 80$ mm. The ground truth backlash models $\mathcal{B}_1(\cdot)$ and $\mathcal{B}_2(\cdot)$ both have a variable backlash width of approximately 1 mm (see green curves in Figure 3.7).

Five minutes of periodic actuator motions, of amplitude chosen to maintain the tracked landmark in the image, are simulated and sampled at $T_s = 0.04$ s. From q_k, the ground truth c_k is obtained by using the ground truth backlash models (see Figure 3.6). From c_k and $p_{1,k/\mathcal{F}_b}$, the image measurements z_k are simulated as defined by Equation (3.10) with an additive measurement noise v_k whose standard deviation is 1 pixel. The FM-DEKF filter is implemented with hyperparameters $R_k = (4)^2 I_{2\times 2}$, $Q_k = 0$, and $\alpha = 2.10^{-3}$ (equivalent to a decay time-constant of 20 s). The state estimate and covariance are initialized with

$$\sqrt{\mathrm{diag}(\mathrm{Var}(\hat{c}_0))} = 1.10^{-3} \begin{bmatrix} \Delta c_1 & \Delta c_2 \end{bmatrix}$$
$$\sqrt{\mathrm{diag}(\mathrm{Var}(\hat{\theta}_0))} = 1.10^{-3} \begin{bmatrix} \frac{\Delta c_1}{\Delta q_1} & \Delta q_1 & \frac{\Delta c_2}{\Delta q_2} & \Delta q_1 \end{bmatrix}$$
$$\sqrt{\mathrm{Var}(\hat{p}^z_{1,0/\mathcal{F}_c})} = 20 \text{ mm}, \quad \hat{p}^z_{1,0/\mathcal{F}_c} = 200 \text{ mm}$$

where Δc_i and Δq_i are the i^{th} distal cable and actuator expected ranges (all set to 6.6 mm) introduced to normalize the initial covariance terms.

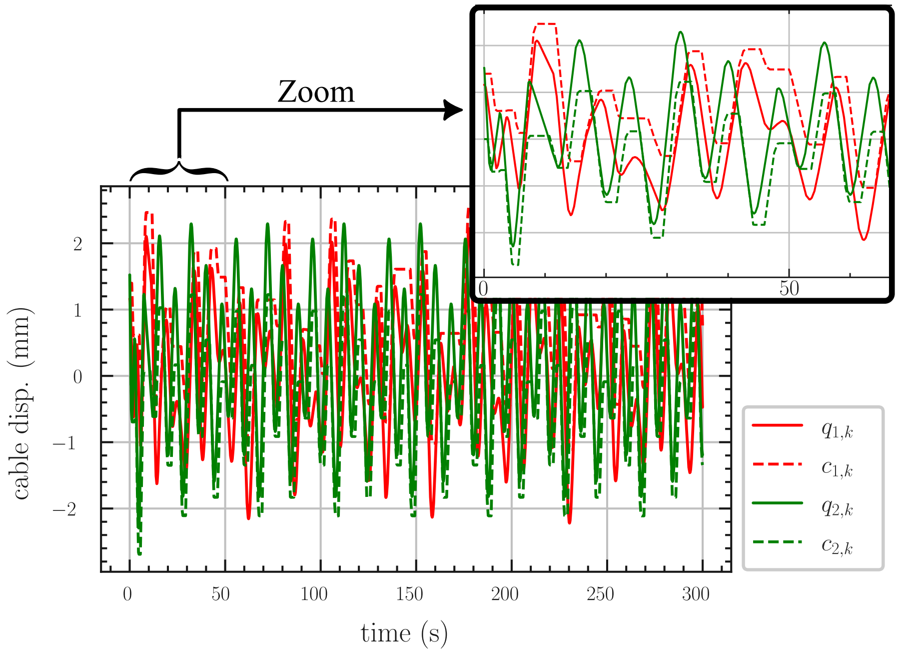


Figure 3.6: Input actuator positions $q_{i,k}$ and backlash element outputs $c_{i,k}$ in mm for each DOF.

3.3.2 Backlash width estimation when $\hat{m}_{O,i}$ is correct

First, we consider the scenario where $\hat{m}_{O,i}$ and $\hat{C}_{O,i}$ have been identified during an offline procedure, but the values for $\hat{C}_{O,i}$ are incorrect once *in-situ*. This is a realistic scenario because the $\hat{C}_{O,i}$ values are dependent from the configuration of the endoscope body (including the passive flexible section) that can change during its insertion (Ott et al. 2011). The estimation errors on $C_{O,i}$ are -0.63 mm and 0.54 mm on the first and second DOF respectively, which is very significant considering that the simulated backlash widths are around 1 mm for both DOF. The mean slopes are set to $\hat{m}_{O,1} = m_{O,1} = 1.1$ and $\hat{m}_{O,2} = m_{O,2} = 1.15$.

Because the $\hat{C}_{O,i}$ values are incorrect, the learned overall backlash model is incorrect as can be observed in Figure 3.7 where the real and estimated hysteretic cycles are reported. However, the backlash width itself is correctly estimated such that the estimated cycle is identical to the real one, provided a translation along $c_{i,k}$ (see Figure 3.7). This is evidenced by the fact that $\hat{\Gamma}^B_{i,k}$ and $\hat{\gamma}^B_{i,k}$ (reconstructed from the estimated parameters $\hat{\Gamma}^A_{i,k}$ and $\hat{\gamma}^A_{i,k}$) converge towards the ground truth values (see Figure 3.8, top). Although both $\hat{\gamma}^B_{i,k}$ values converge more slowly than the estimated parameters $\hat{\Gamma}^B_{i,k}$, all parameters have converged after three minutes of simulated motion (see Figure 3.8, top). There is a small residual error on the $\hat{\gamma}^B_{i,k}$ values caused by the incorrect fixed parameters $\hat{C}_{O,i}$, but there is no significant error on the resulting estimated backlash width model (see Figure 3.8, bottom). In Figure 3.8, one can also note that the backlash width modeling errors are already very significantly reduced after two minutes.

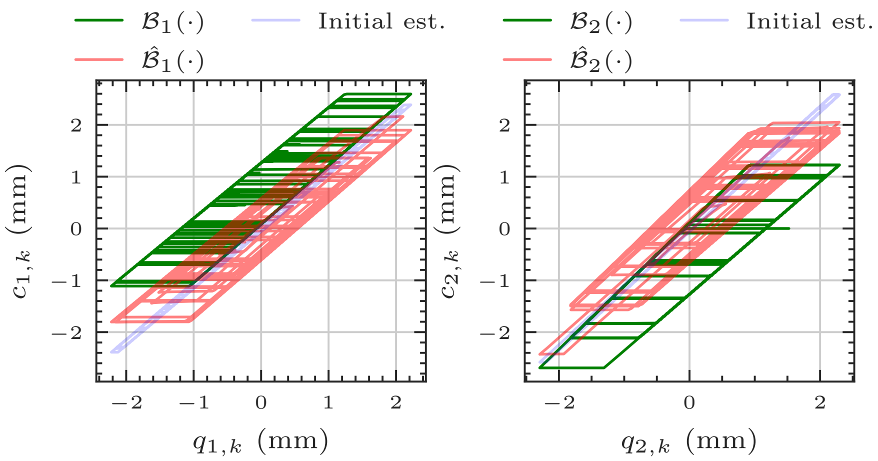


Figure 3.7: Hysteretic cycles resulting from the ground truth (green) and estimation of c_k (red) during the learning process in the case where the estimates of $m_{O,i}$ are correct. The initial model is also represented (blue).

3.3.3 Robustness to incorrect $\hat{m}_{O,i}$ values

In practice, model identification is complex for this type of robot and the fixed parameters $\hat{m}_{O,i}$ could be slightly inaccurate. Consequently, both the $\hat{m}_{O,i}$ and $\hat{C}_{O,i}$ parameters may – and probably will – be inaccurate in any real scenario. Under the same conditions as in the previous simulation, an error is introduced on the estimated parameters $\hat{m}_{O,i}$ such that $\hat{m}_{O,1} = \hat{m}_{O,2} = 1$ (i.e., estimation errors are respectively 10% and 15%). The learning process results in performances comparable to those obtained with correct $\hat{m}_{O,i}$ values, with a convergence of the parameters towards their correct values in three minutes (see Figure 3.9, bottom). Similarly, although the final $\hat{\Gamma}^B_{i,k}$ estimates are very accurate ($< 2\%$ errors), there are small residual errors on the $\hat{\gamma}^B_{i,k}$ estimates ($\approx 10\%$).

3.3.4 Discussion

The proposed method is capable of learning the correct parameters of the backlash width model even if the fixed parameters $C_{O,i}$ and $m_{O,i}$ are inaccurate. The residual parameter estimation errors, resulting from the misspecification of the fixed parameters, only incur small errors on the estimated backlash width $\hat{b}_i(\cdot)$: under 0.02 mm, i.e., less than 2% of the mean backlash widths. The seemingly long convergence time (i.e., 3 minutes) is partially explained by the nature of the learning method. The FM-DEKF filter only updates the parameters when they are identifiable and, therefore, does not update them when the associated DOF is in the deadzone. The simulated backlash width is quite large when compared to the amplitude of the motions and, as a result, the two actuators are respectively 53% and 44% of the time in the deadzone during the learning phase. This effectively reduces by two the amount of data used for learning, which results in a longer identification procedure. A more aggressive parameter tuning could increase the learning rate, but at the risk of reducing robustness in a real life scenario. Additionally, it should be noted that since we essentially perform a local identification around a given

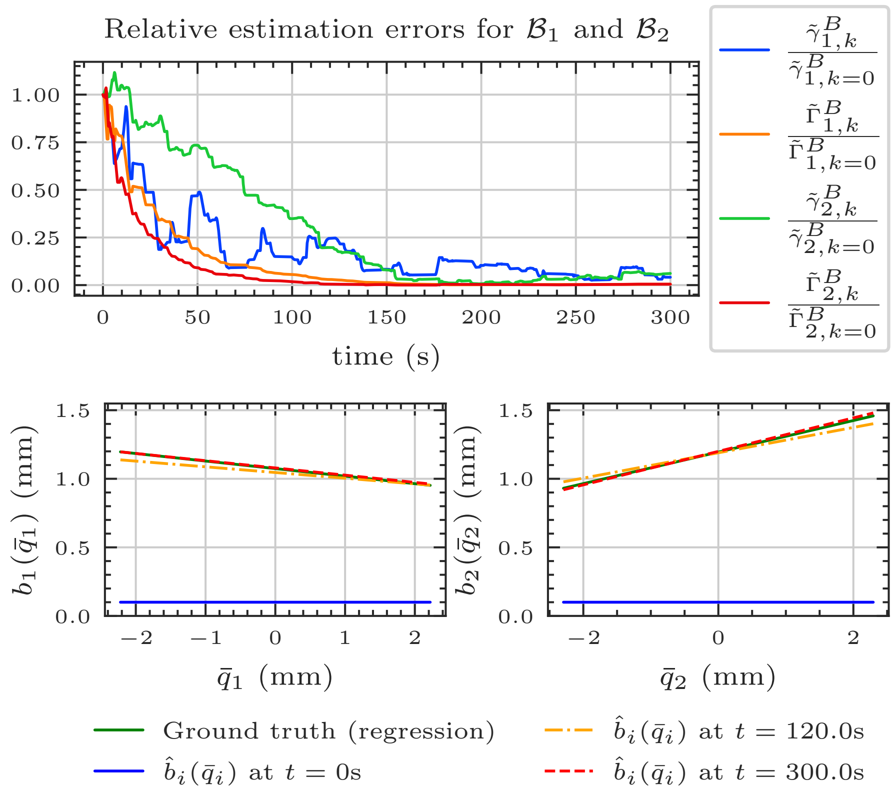


Figure 3.8: Top: relative backlash model parameter estimation errors in the case where the estimates of $m_{O,i}$ are correct. The relative errors are computed as the ratio of the current estimation error (e.g., $\tilde{\gamma}^B_{1,k} = \gamma^B_{1,k} - \hat{\gamma}^B_{1,k}$) by the initial error. Bottom: Resulting estimated model $\hat{b}_i(\bar{q}_i)$ and ground truth.

robot pose, the model can be simplified by setting $\gamma^A_{i,k} = \gamma^B_{i,k} = 0$ if the backlash width is, at least locally, constant.

Although the scene reconstruction is not of interest here, it is worth commenting on the estimated landmark position that is a by-product of the identification process. When $\hat{C}_{O,i}$ is incorrect (see Section 3.3.2), the estimated landmark position in the endoscope base frame $\hat{p}_{1,k/\mathcal{F}_b}$ cannot converge towards the correct value. However, the estimated landmark position w.r.t. the camera frame $\mathcal{F}_c$ is correct (see Figure 3.10, top). This is explained by the fact that the transformation ${}^bT_c$ (see Equation 3.2) depends on c_k and that $\hat{c}_k$ itself will be inaccurate if $\hat{C}_{O,i}$ is incorrect (i.e., static offset). Finally, if both $\hat{C}_{O,i}$ and $\hat{m}_{O,i}$ values are incorrect, then the landmark estimated position is incorrect regardless of the frame of reference (see Figure 3.10, bottom).

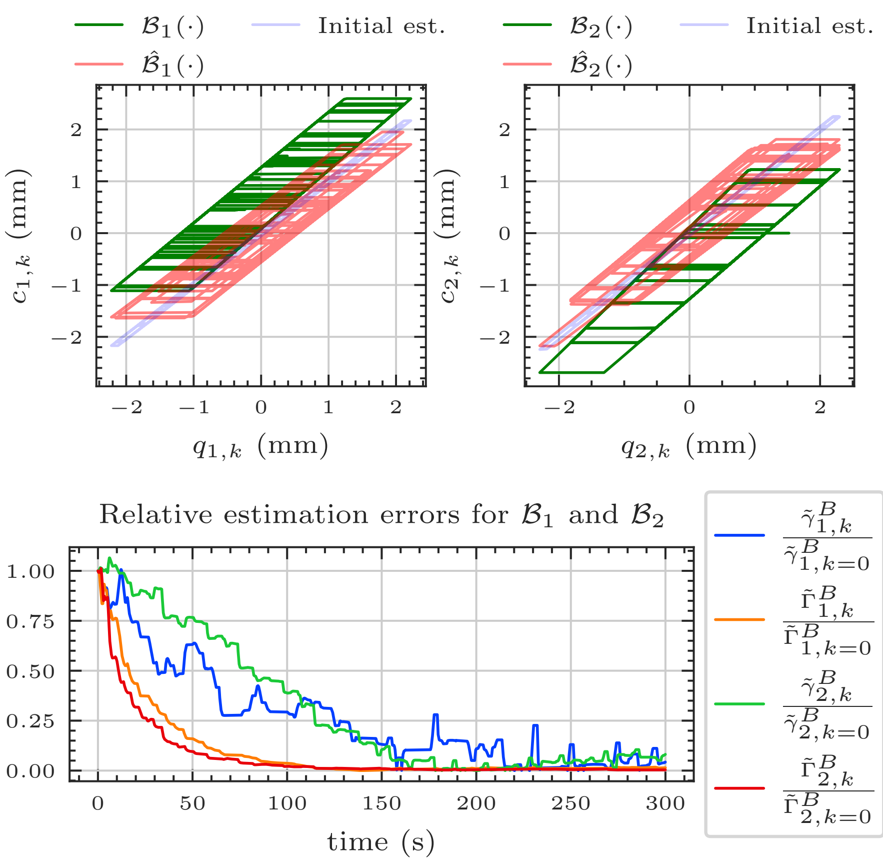


Figure 3.9: Top: hysteretic cycles resulting from the ground truth and estimation of c_k during the learning process in the case where $m_{O,i}$ and $C_{O,i}$ estimates are incorrect. Bottom: relative backlash model parameter estimation errors in the case where all $m_{O,i}$ and $C_{O,i}$ estimates are incorrect.

3.4 Experimentation on a robotic endoscopic platform

3.4.1 Experimental setup and ground truth estimation

Experimental validation was carried out on a flexible endoscope[1] whose two bending DOF are robotized. The test bed is composed of the endoscope, an environment containing an ArUco marker visible in the endoscopic image, and a joystick used for manual teleoperation (see Figure 3.11). The endoscopic images of resolution 720×576 are acquired at a 25 Hz frame rate ($T_s = 0.04$ s) and synchronized with the motor positions measurements. A data post-treatment is implemented to remove image distortion (camera calibrated beforehand), track the marker, and compensate for the image latency resulting from the image acquisition pipeline (evaluated at $2T_s$).

A calibration sequence is first executed (twice) where the endoscope automatically performs slow DOF by DOF motions resulting in the image displacements reported in Figure 3.12 (left). The backlash width is manually measured from the figure for

[1]Gastroscope, model 13806PKS from Karl Storz, Germany.

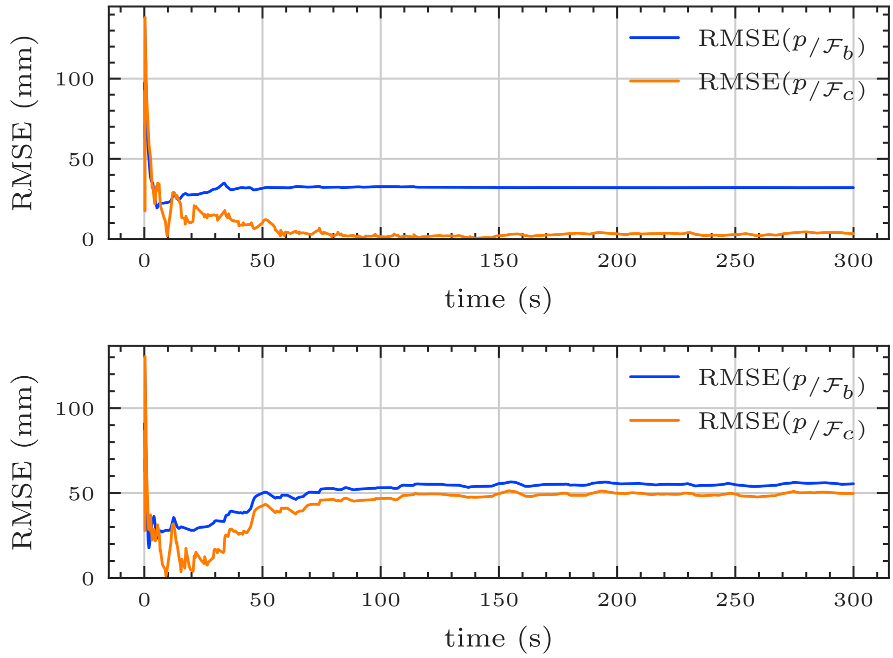


Figure 3.10: Top: RMSE of the landmark position estimation in the case where the estimates of $m_{O,i}$ are correct. Bottom: Case where the $m_{O,i}$ and $C_{O,i}$ estimates are incorrect.

different mean actuator positions $\bar{q}_i$ (see Figure 3.12, right). This process is similar to the one proposed by Ott et al. (2011), except that in our case it is the width between the two envelope functions that is measured and not the pure backlash occurring when the deadzone is crossed. The ground truth parameters and standard errors are then retrieved by linear regression on the data collected during the two validation sequences (also reported in Figure 3.14, bottom):

$$\Gamma_1^B = 1.2 \pm 0.01 \text{ (mm)}, \quad \gamma_1^B = 0.02 \pm 0.01 \text{ (unitless)}$$
$$\Gamma_2^B = 0.61 \pm 0.02 \text{ (mm)}, \quad \gamma_2^B = -0.1 \pm 0.02 \text{ (unitless)}$$

Two minutes of teleoperated motions are then realized and the learning method is applied using the same hyperparameters and initial conditions than for the simulation

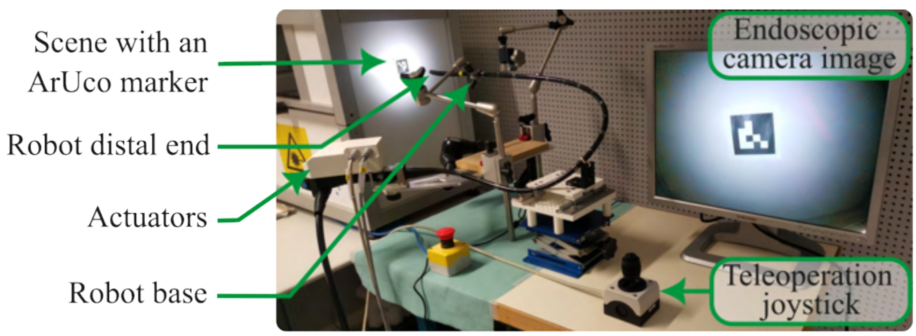


Figure 3.11: Experimental test bed with a robotized endoscope.

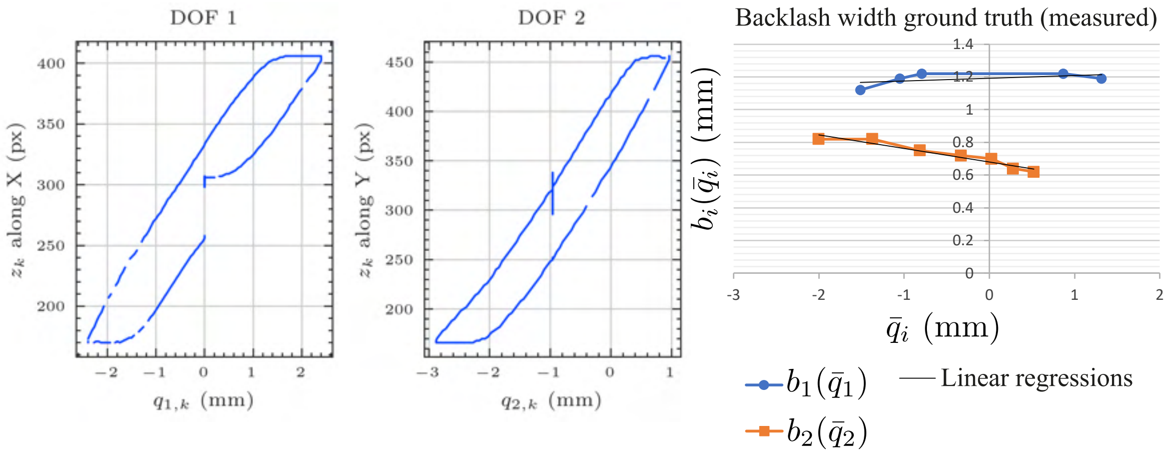


Figure 3.12: Left: image measurements acquired during a calibration sequence composed of slow, smooth DOF by DOF motions. Right: manual backlash width measurement from calibration sequence and resulting linear regression.

presented in Section 3.3. The image trajectory is reported in Figure 3.13 and shows the arbitrary teleoperated motions of the endoscope. The fixed model parameters are $\hat{C}_{O,1} = \hat{C}_{O,2} = 0$, $\hat{m}_{O,1} = 0.8$ and $\hat{m}_{O,2} = 1$, the last two coarsely identified empirically by ascertaining that equal actuator velocities $\dot{q}_{1,k}$ and $\dot{q}_{2,k}$ result in a similar image velocity $\|\dot{z}_k\|$.

3.4.2 Results

After two minutes of learning, the parameter estimation errors are $\tilde{\Gamma}^B_{1,k} = \Gamma^B_{1,k} - \hat{\Gamma}^B_{1,k} = 0.05$ mm, $\tilde{\gamma}^B_{1,k} = -0.003$, $\tilde{\Gamma}^B_{2,k} = 0.04$ (unitless), and $\tilde{\gamma}^B_{2,k} = -0.02$ (see Figure 3.14, top). The parameter $\hat{\gamma}^B_{1,k}$ happens to be already correct at startup (i.e., near constant backlash width on this DOF) and varies little overall. The other backlash width model parameters errors are significantly reduced over the identification procedure: by more than 90% for $\hat{\Gamma}^B_{1,k}$ and $\hat{\Gamma}^B_{2,k}$, and by 80% for $\hat{\gamma}^B_{2,k}$ (see Figure 3.14, top). At the end of the identification process, the mean backlash width estimation error $|b_i(\bar{q}_i) - \hat{b}_i(\bar{q}_i)|$ over the range of actuator positions is under 0.1 mm for both DOF (see Figure 3.14, bottom). The backlash width estimation errors are then reduced by 95% and 90% for DOF 1 and 2, respectively, when compared to the initial model.

3.4.3 Discussion

Experimental results showed that the backlash widths could be correctly estimated during the teleoperation of the endoscope, hence without the need for specific DOF-by-DOF motions as in (Ott et al. 2011) or (Poignonec et al. 2020). This means that the identification could be achieved during normal operational motions of the endoscopes, which opens perspectives for continuous backlash learning during contactless surgeries such as laser dissections. Furthermore, as the learning is performed online, the method could potentially cope with *in-situ* variations of the endoscope configuration, including

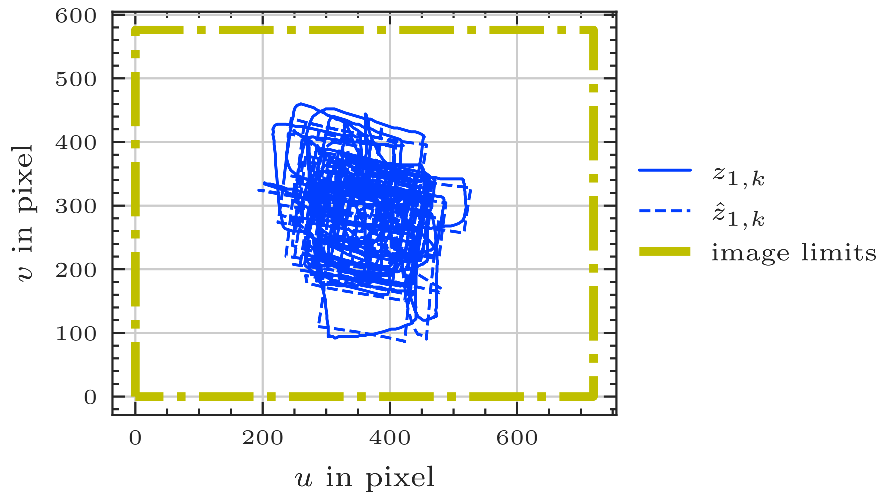


Figure 3.13: Image measurements collected during the experiments and (posterior) prediction by the FM-DEKF filter.

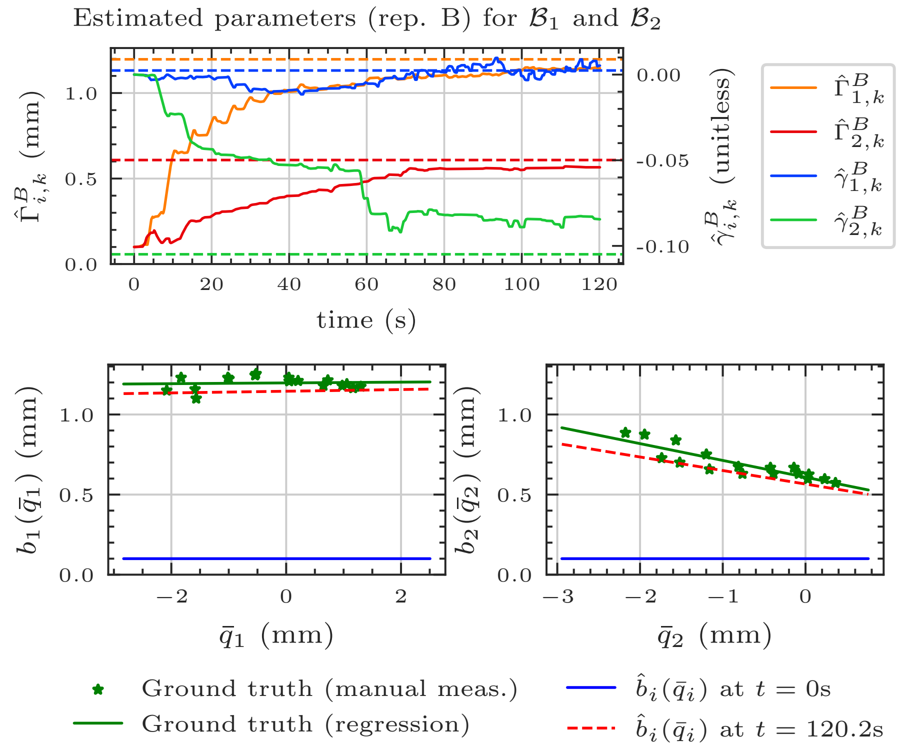


Figure 3.14: Top: estimated backlash model parameters and ground truth estimate for the experiment with a real endoscope. Ground truth values are represented with dotted lines. Bottom: Resulting estimated model $\hat{b}_i(\bar{q}_i)$ and ground truth.

the passive flexible body.

There is a larger estimation error on $\hat{\gamma}_{2,k}^{B}$, but this error is comparable to the uncertainty (i.e., standard error) on the ground truth estimate. An explanation for this uncertainty could be that the backlash model is affected by the velocity, but that would not explain the uncertainty of the ground truth values since the velocity for both validation sequences were similar. A second explanation is that artifacts are due to non-linear transitions in the real hysteretic model of the robot as was investigated by Ott et al. (2011). Such non-linear effects could also explain why the backlash width is underestimated (see Figure 3.14): due to non-linear transitions, the system leaves the deadzones earlier than it would if the backlash was as modeled in Section 3.1.2. Therefore, modeling these transitions could be beneficial, e.g., by using a Bouc-Wen model (Chatzis et al. 2017b).

3.5 Conclusions

An automatic backlash width identification method applicable to eye-in-hand flexible endoscopes was proposed. The main contribution lies in the novel problem formulation, notably the separation of backlash width from the overall hysteretic model. An online implementation based on discontinuous extended Kalman filtering was proposed and validated, but other implementations of the approach could be investigated, e.g., methods based on structure from motion. Simulation results showed that the correct backlash width model parameters could be learned from endoscopic images and that the proposed method was robust to modeling errors on the model parameters not updated online. Experimental results on an endoscopic platform were also reported, showing that the variable backlash width could be estimated *in-situ* with small residual errors (i.e., about 10% of the real width). Although it is out of scope here, it should be noted that an online identification makes the simultaneous identification and compensation of backlash possible. Compensating the backlash during the identification would allow to reduce the time spent in actuator deadzones, which could in turn reduce the identification time.

Some limitations of the method as presented lie in the management of the tracked features map. Firstly, experimenting with more features and different scenes (e.g., topology, distance) would be interesting. Secondly, the tracked landmark was voluntary kept within the image limits (see Fig 3.13), but if the landmark(s) are very close to the camera, they might disappear such that a map of the environment will have to be managed, e.g., as in (Grasa et al. 2014). Not having to keep landmarks within the image bounds would also allow the generation of predefined automatic motion, ideally chosen to maximize the information available for learning.

It should also be noted that both the endoscope and the scene were static during the experiments, but it may not be so in an in-vivo setting. An interesting perspective of this work would consist in including the physiological motion in the learning process, for instance by introducing a parametric model of the physiological motion similarly to what is done in (Mountney et al. 2010).

Chapter 4

Backlash width identification from motion detection events

Contents

4.1 Modeling of the backlash as cascaded non-linearities 55
4.2 Estimation of the pure backlash component 56
4.2.1 Estimation of the backlash at a given motor position 56
4.2.2 Construction of the backlash width function 57
4.2.3 Use for backlash compensation 58
4.3 Remaining function reconstruction 59
4.3.1 General method 59
4.3.2 Proposed implementation for 1 DOF identification 59
4.3.3 Use for improved robot positioning 60
4.4 Experimentation on the STRAS endoscopic platform 60
4.4.1 Experimental setup 60
4.4.2 Method implementation 61
4.4.3 Results 62
4.4.4 Discussion 64
4.5 Conclusions 65

In the last chapter, we demonstrated that the backlash width can be learned independently from the overall hysteretic relation between proximal and distal displacements. Such separation of the backlash properties allowed to relax the necessary conditions for parameter identifiability, but a model of the endoscope and accurate measurements were still critical to perform the backlash width identification. However, in principle, the pose of the robot is of little consequence if only the backlash width is learned, since the backlash width model only predicts whether an actuator is in the deadzone or not, given the history of displacements. Then, we argue that, in some conditions, the backlash width could be learned from a unique easily acquired measurement, such as a binary signal encoding the presence or the absence of distal motion.

In this chapter, we propose a novel backlash estimation method adapted to the case of eye-to-hand endoscopes. The proposed approach is complementary to methods based on direct (see Chapter 2) or indirect (see Chapter 3) robot pose estimation. First, we formulate the backlash estimation problem as two distinct sub-problems: the estimation of the backlash width and the estimation of the remaining function. This leads to a cascaded backlash as represented in Figure 4.1 where the two components can be learned separately. Then, we propose a motion detection-based approach to learn the backlash width referred to as the pure backlash component. The absence or presence of motion are events directly visible in the endoscopic image. The acquisition of this information does not require a 3D pose estimation process and, therefore, neither does it need a model of the robot, of the camera, or any other kind of registration or kinematic information. Finally, we propose a method based on pose estimation to recover the remaining information that cannot be learned from motion detection alone. This process is independent from the backlash width estimation, such that if the pose cannot be reconstructed (e.g., because of insufficient measurements or lack of kinematic model), the backlash width model can still be learned *in-situ*. The full identification process is evaluated experimentally on a cable-actuated flexible tool of the STRAS endoscopic platform (Zorn et al. 2018) displayed in Figure 1.4c.

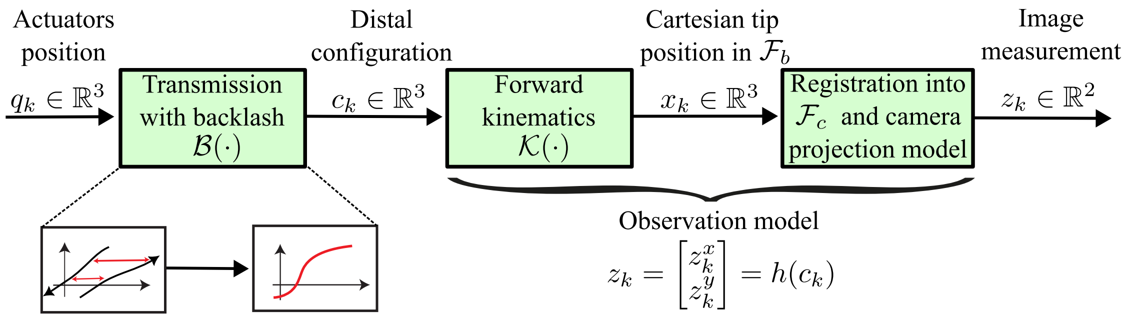


Figure 4.1: Proposed modeling of the continuum robot, where transmission nonlinearities are split in two parts. Note that in practical applications, measurements z_k are sampled at a significantly lower rate that q_k.

4.1 Modeling of the backlash as cascaded non-linearities

Let us consider the i^{th} DOF of a tendon-driven continuum robot whose non-linearity is expressed as an hysteretic relation from proximal actuation $q_{i,k}$ to distal joint or configuration variable $c_{i,k}$. As a reminder, in chapters 2 and 3, an analytical model with switching conditions of the form $c_{i,k} = \mathcal{B}_i(\theta_i, c_{i,k-1}, q_{i,k})$ was introduced to model this relation. Instead, let the backlash model be a combination of two cascaded elements as illustrated in Figure 4.1: a “pure” backlash operator followed by a static non-linearity, similarly to what is done by Rochdi et al. (2010) or Vörös (2018). These two components are as follows:

- the pure backlash corresponds to a deadzone of possibly variable width appearing at a change of direction of the actuator. Using the notations from Chapter 3, it is defined as a backlash operator for which $c_{i,k} = \bar{q}_{i,k}$[1] and whose backlash width $b(\bar{q}_i)$ is an arbitrary function (see Figure 4.2a-b); The pure backlash is then necessarily symmetric such that, for a given distal position, the deadzone to “cross” spans the same motor range, regardless of the direction of the motor motion. Intuitively, the pure backlash is obtained by deforming the real hysteretic cycle such that across the whole motor range, the deadzone is centered on the $c_{i,k} = q_{i,k}$ characteristics (see Figure 4.2b)
- the possibly non-linear function $f_{remain}(q_i)$ links the actuator position to the configuration variable value once pure backlash has been removed or compensated as illustrated in Figure 4.2c. In the following, we call this function the remaining function. This function can also be viewed as the central line of the hysteresis curve (see Figure 4.2a). It should be noted that it is generally different from the envelope functions of the hysteresis (see for instance Figure 4.4).

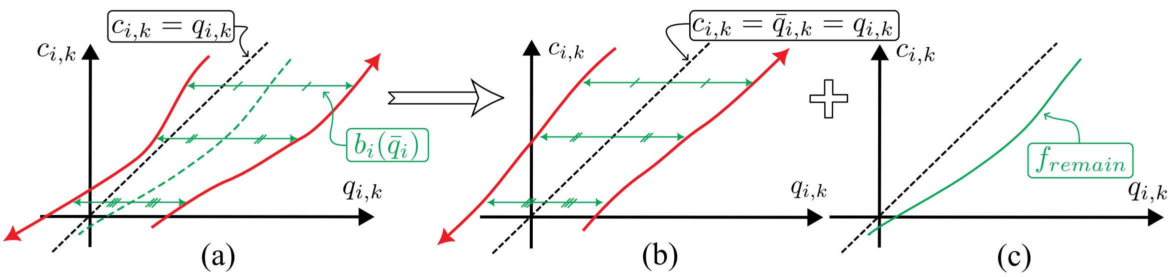


Figure 4.2: The backlash (a) is decomposed in a pure backlash component (b) and a remaining function that is possibly non-linear (c).

[1] $\bar{q}_{i,k}$ is the mean actuator position among those that could result in a given output $c_{i,k}$. This notion is introduced in Chapter 3 (see Equation 3.4).

4.2 Estimation of the pure backlash component

4.2.1 Estimation of the backlash at a given motor position

Pure backlash can be characterized by the absence of end-effector motion after a change of direction of the actuator at the proximal side. The backlash width is then defined as the range of motor motion which does not create distal motion. The proposed approach therefore consists in detecting the appearance of motion after a change of motor direction. Distinguishing robot distal motion in the image resulting from one actuator displacement or another would require very accurate models, which is not desirable here. Therefore, the backlash estimation is performed for only one DOF at a time, such that any displacement detected in the image is necessarily caused by the commanded actuator motion. The continuous time notation used in the following assumes that the motor positions are obtained at a high frame rate with negligible delay, which is usually the case in mechatronic systems. The process for (local) backlash estimation is illustrated in Figure 4.3 and can be described by the following steps:

- the position of the effector is tracked in the image and the motor position are recorded;
- an actuator direction change (change of $\text{sign}(\dot{q}_i)$) is requested at time t^-;
- a movement is detected in the image at time t^+;
- the local backlash width is computed from the recorded motor positions.

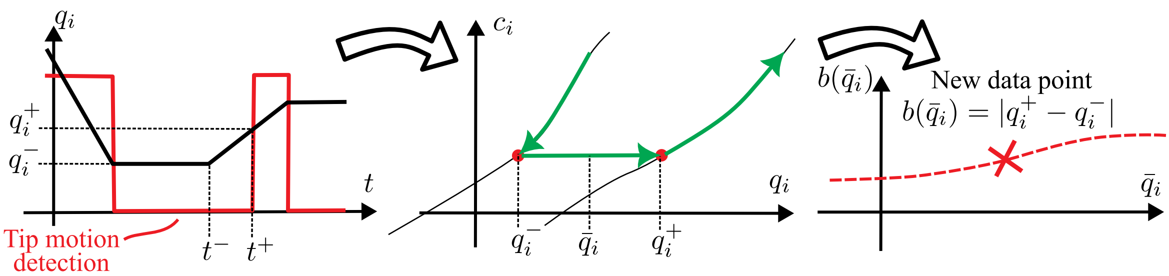


Figure 4.3: Pure backlash estimation scheme. Left: actuator motions for a requested change of direction, and the detected motion in the image. Center: corresponding evolution in the configuration vs actuator space. Right: The measured backlash width provides a data point in the backlash width function.

As discussed in Chapter 3, the backlash width can be considered to be a function of the distal configuration c_i. However, since c_i is not directly accessible (i.e., no distal measurements), its estimation would require a correct model of the kinematics and a pose estimation process. Therefore, it is more convenient to describe the backlash width as a function of an actuator position, which is readily available. We express the backlash width as a function of the mean actuator position in the backlash zone denoted as $\bar{q}_i$ as was done in Chapter 3 and illustrated in Figure 4.3 (centre). In the absence of time delays, the local value of the backlash width $b_i(\bar{q}_i)$ is then given by

$$\begin{cases} b_i(\bar{q}_i) = \left| q_i(t^+) - q_i(t^-) \right| \\ \bar{q}_i = \dfrac{q_i(t^-) + q_i(t^+)}{2} \end{cases} \tag{4.1}$$

It can represent backlash for both direction changes, either positive to negative or negative to positive velocity. However, since different sources of information are used to obtain t^- and t^+ a correction is needed to temporally align image measurements and motor measurements. This is especially important when using low frame rate image acquisition systems, which also introduce delays. Consequently, we use the following relations instead:

$$\begin{cases} b_i(\bar{q}_i) = \left| q_i(t^+ - t_{delay}) - q_i(t^-) \right| \\ \bar{q}_i = \dfrac{q_i(t^-) + q_i(t^+ - t_{delay})}{2} \end{cases} \tag{4.2}$$

where t_{delay} is the motion detection delay introduced by the image acquisition and processing pipeline.

4.2.2 Construction of the backlash width function

As the backlash width depends on q_i, the method presented in Section 4.2.1 must be performed at different actuators positions, in order to estimate the function $b_i(\bar{q}_i)$ over a useful range of actuators positions. For an *in situ* estimation prior to use, typically after the medical robot insertion in the body, but before the actual surgical use, a pre-programmed motor trajectory can be used to reconstruct the backlash width function $b_i(\bar{q}_i)$. This trajectory can consist in several small range back and forth motions superimposed on a ramp which covers the full range of the useful motor positions. Interestingly, since the sensor used for the backlash estimation (the endoscopic camera) is the one which is also used in practice[2] (e.g., visual feedback during teleoperation), the useful range of values is actually the one where measurements are possible. Using this pre-programmed trajectory, one can estimate values of $b_i(\bar{q}_i)$ over the desired range, but only at discrete intervals. Using the backlash model for compensation (see Section 4.2.3) requires, however, a continuous function for all motor positions. We propose to interpolate and filter the obtained values by using cubic B-splines, with N control points homogeneously spread over the motor range. This choice was made based on the observation that some of the hystereses obtained with external sensors exhibit an overall smooth variation, but with possible high local slopes (Aleluia Porto et al. 2019).

Given an initial coarse model of $c_{i,k} = f_{remain}(q_{i,k})$ denoted $f_{coarse}(q_{i,k})$, which can be obtained from the kinematic modeling of the tool, the direct (complete) backlash model can be obtained by combining the backlash model and the f_{coarse} function. Considering a decreasing f_{coarse} [3] , the external envelope defined by functions f_L and f_R can be

[2]In the case of eye-to-hand endoscopes.

[3]The following is valid for deacreasing f_{coarse} and f_{remain} functions, but also valid for increasing function provided some (minor) changements (i.e., signs in equations mostly). The choice of decrease for explanations is motivated by the actual experimental data of the considered robot.

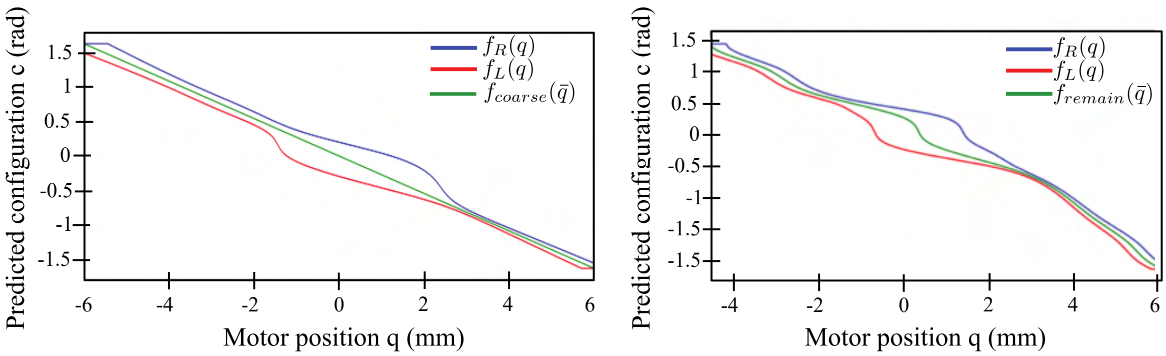


Figure 4.4: Reconstructed hystereses, by combining a linear f_{coarse} function with pure backlash (top) and by combining estimated f_{remain} with backlash (bottom).

obtained by a numerical reindexing of f_{coarse} such that

$$\begin{cases} f_R(q_i) = f_{coarse}(\bar{q}_i), \text{ with } q_i = \bar{q}_i + \dfrac{b_i(\bar{q}_i)}{2} \\ f_L(q_i) = f_{coarse}(\bar{q}_i), \text{ with } q_i = \bar{q}_i - \dfrac{b_i(\bar{q}_i)}{2} \end{cases} \tag{4.3}$$

This is illustrated in Figure 4.4 (left), where f_{coarse} is linear. Note that because the backlash width depends on q_i, $f_{L,R}$ are generally not linear since the variable backlash model enforces a deformation of the external envelope.
At a time index k the direct model can be numerically simulated by

$$c_{i,k} = \begin{cases} f_R(q_{i,k}), \text{ if } q_{i,k} > q_{i,k-1} \text{ and } c_{i,k} \geq f_R(q_{i,k}) \\ f_L(q_{i,k}), \text{ if } q_{i,k} < q_{i,k-1} \text{ and } c_{i,k} \leq f_L(q_{i,k}) \\ c_{i,k-1}, \text{ otherwise} \end{cases} \tag{4.4}$$

4.2.3 Use for backlash compensation

The identified backlash width model can be used to compensate the backlash. Given a reference trajectory $c^r_{i,k}$ for the configuration variable, the compensated reference motor trajectory can be computed as:

$$q^r_{i,k} = \begin{cases} F_R(c^r_{i,k}), \text{ if } c^r_{i,k} > c^r_{i,k-1} \\ F_L(c^r_{i,k}), \text{ if } c^r_{i,k} < c^r_{i,k-1} \\ q^r_{i,k-1}, \text{ otherwise} \end{cases} \tag{4.5}$$

where

$$\begin{cases} F_R(c^r_i) = f^{-1}_{coarse}(c^r_i) + \dfrac{b_i(f^{-1}_{coarse}(c^r_i))}{2} \\ F_L(c^r_i) = f^{-1}_{coarse}(c^r_i) - \dfrac{b_i(f^{-1}_{coarse}(c^r_i))}{2} \end{cases} \tag{4.6}$$

f^{-1}_{coarse} is the inverse function of f_{coarse}, which can be obtained by a numerical inversion when no analytical form exists. Smoothing can also be added to transitions to avoid actuator saturation (Ott et al. 2011).

This compensation can advantageously be used in a telemanipulation framework, for correcting the behavior of the robot when a change of direction of the distal configuration variable is required. It can thus avoid apparent delays in executing the user's reference trajectories.

4.3 Remaining function reconstruction

4.3.1 General method

For automatic motions realization, as described by Aleluia Porto et al. (2019), the compensation of backlash alone is not sufficient for providing a good positioning accuracy. Indeed, f_{coarse} is usually inaccurate, and therefore $f_{R,L}$ are also. There are for instance dead zones near the straight configuration that are difficult to predict from models. This means that even if backlash may be correctly compensated when changes of direction are required, the link between the rate of variation of $c^r_{i,k}$ and of $q^r_{i,k}$ can be incorrect. In other words, the task space position can be inaccurate. For improving accuracy, one can estimate the complete hysteresis envelope, i.e., f_R and f_L, from data. These functions link the actuator positions to the configuration variable values as long as no change of direction is applied. This information can be obtained from the endoscopic camera only if the configuration variable can be extracted from the image. This in turn requires a geometric model of the continuum robot, a model of the camera and of the eye-to-hand configuration. Such techniques have been used to reconstruct three (Reilink et al. 2013a) or six configuration variables (Cabras et al. 2017).

4.3.2 Proposed implementation for 1 DOF identification

In the considered case where a single DOF has to be estimated, a marker at the tip of the instrument coupled with a conventional geometric model (e.g., a constant curvature model) is sufficient. We propose an approach similar to (Reilink et al. 2013a), but instead of reconstructing a corrected motor position, we focus on the configuration variable c (both are equivalent since the underlying model is known). Given a model of the endoscopic camera and of the eye-to-hand calibration, we have

$$\hat{z}_k = \begin{bmatrix} \hat{z}^x_k \\ \hat{z}^y_k \end{bmatrix} = h(c_k) \tag{4.7}$$

Given a position of the marker in the image z_k, one can then estimate the optimal value of c_k by minimizing the cost function $\mathcal{L}(c) = \|z_k - h(c)\|^2$ by using an iterative optimization process such as Levenberg-Marquardt. Note that this approach, as the ones used in (Cabras et al. 2017; Reilink et al. 2013a), minimizes the error of a projection in the image. This scheme can therefore succeed and provide very low projection errors even when models are erroneous. In such cases, the obtained values for the configuration variables will be inaccurate.

By applying this estimation process during the programmed motion of the instrument used for pure backlash estimation as described in Section 4.2.2, one can reconstruct the hysteresis characteristic curves for the DOF of interest (see Figure 4.6b). After this step, the functions defining the external envelope of the hysteresis could be used to compute the width of the backlash in function of $\bar{q}_i$ as for instance in (Aleluia Porto et al. 2019). However, in this case the pure backlash estimation obtained previously (see Section 4.2) would be overwritten. This is not desirable, because contrary to (Aleluia Porto et al. 2019) where accurate external sensors are used, the process used here to estimate c is subject to modeling and measurements errors. We therefore propose to keep the initial pure backlash estimation, which was obtained independently of any pose estimation, and to estimate the remaining function. For this purpose the remaining function is estimated as the central fiber of the characteristics, i.e., the horizontal median line of the envelope. For extracting it, the characteristic is processed as an image by applying morphological operations for filling holes, extremal horizontal value detection and spline fitting. This provides function f_{remain} as shown in Figure 4.6b.

4.3.3 Use for improved robot positioning

The complete obtained model can for instance be used for an automatic positioning task as will be shown in Section 4.4. Given a desired trajectory $c^r_{i,k}$ for the configuration variable, the required actuator position can be obtained using equations (4.5) and (4.6), but replacing f_{coarse} with f_{remain}. The complete estimation scheme is represented in Figure 4.5.

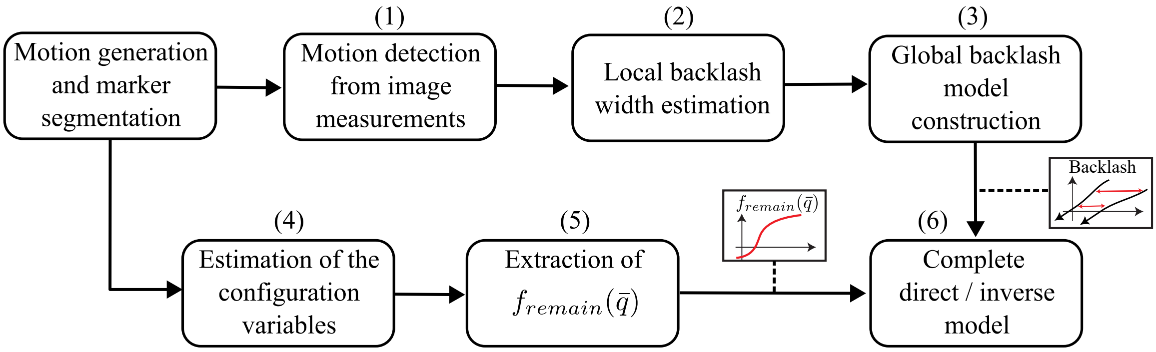


Figure 4.5: General scheme used for non-linearities estimation. The upper part of the block-scheme (blocks 1 to 3) concerns estimation of pure backlash while the lower part (blocks 4,5) processes the remaining function. They are combined to provide the complete estimate (block 6).

4.4 Experimentation on the STRAS endoscopic platform

4.4.1 Experimental setup

We consider the left instrument of the STRAS robotic platform (see Figure 1.4c) visible in Figure 4.9 whose kinematic model was given in Chapter 2 (see Section 3.1). The instrument has three DOF : translation, rotation and bending. The bending DOF actuated by antagonist cables is the main focus of the experiments. The robot DOF are controlled in position by a real-time control PC (see Zorn et al. 2018, for details).

Endoscopic images are acquired at 25 Hz through a Karl Storz Telecam acquisition system. Images are then transferred via S-video connection to a Euresys Piccolo PCIe frame grabber in a host computer where image processing and signal processing are carried out. The processing codes are in C++ and the robot motions are commanded by a network socket connection between the image processing PC and the control PC. Inverse backlash models are implemented on the robot control PC in a layer between joint references (coming from master interfaces or from pre-planned tasks) and the low-level control of the motors.

4.4.2 Method implementation

Image measurements

The continuum robot end effector needs to be tracked in the image in order to reconstruct the configuration variable for the identification of f_{remain}. Since a single DOF is considered, it is sufficient to track a single point. The focus of the present work is not on the image processing aspects, so we used a simple colored marker attached at the tip of the instrument, which can be efficiently tracked using simple image processing techniques. More advanced marker-based (Cabras et al. 2017; Reilink et al. 2013a) or marker-less (Baek et al. 2020; Reiter et al. 2011; Rosa et al. 2019) techniques can be used for in vivo settings.

Backlash estimation

The delay of the image acquisition chain has been estimated to 2 frames (i.e., $t_{delay} =$ 80 ms), with an uncertainty $\Delta t_{delay} = 20$ ms (half period). Assuming a constant velocity of the actuator v during backlash crossing, this creates an estimation error $\Delta q_i = \|\Delta t_{delay} v_i\|$ on $q_i(t^+ - t_{delay})$. It is therefore recommended to apply a low velocity for backlash estimation. Practically, the velocity is set to 5% of the maximum motor velocity, which corresponds approximately to a maximum bending speed of $20°/s$ at the distal end. The corresponding uncertainty onto the actuator position is 0.025 mm of cable displacement. For separating motion and absence of motion, the image position of the marker z_k is differentiated and a threshold is applied. This threshold was set to 0.2 pixels/frame after practically assessing that the noise for still marker localization was less than 0.15 pixels.

Figure 4.6a shows the backlash estimation as a function of the motor position. The continuous function has been obtained by B-spline fitting with $N = 14$ knots (8 central and 3 at each extremity of the motor range). One clearly observes the strong increase of the backlash near the center of the motor range (straight configuration of the instrument). This behavior was already observed in (Bardou et al. 2012; Aleluia Porto et al. 2019) and is linked to the need for re-tensioning the cable after a change of direction added to the simultaneous slacking of both cables near the straight configuration of the instrument. This clearly confirms the need of a variable backlash estimation method (see Section 4.1).

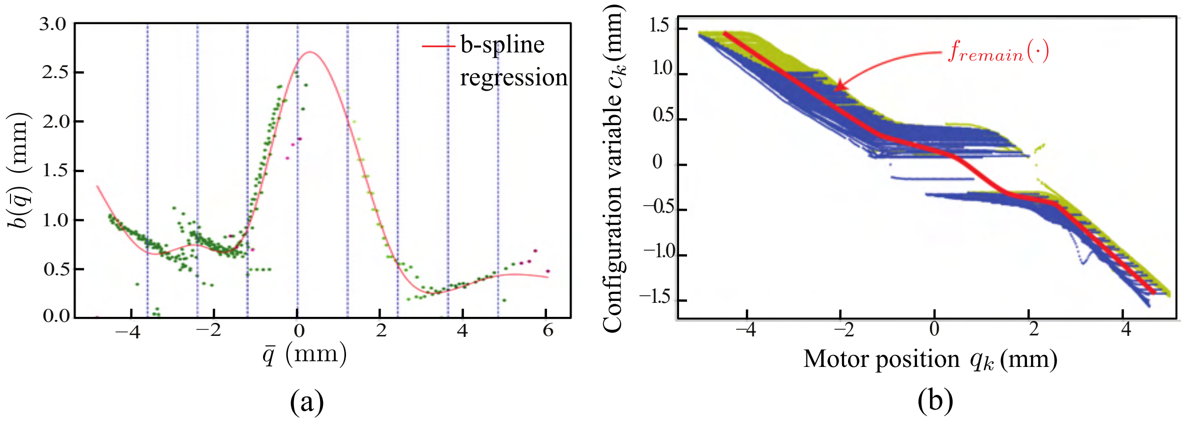


Figure 4.6: (a) Backlash width in mm of cable displacement as a function of the mean actuator position. Blue vertical lines indicate the positions of the spline knots. (b) Characteristics between the actuator and configuration variable obtained using configuration variable estimation from images.

Remaining function estimation

The remaining function was estimated using the same input images and motor data as for the backlash width. The typical actuator / configuration characteristics are shown in Figure 4.6b. f_{remain} is then extracted as described in Section 4.3.

4.4.3 Results

Assessment of prediction capability

For assessing the validity of the pure backlash and the remaining function estimation, a testing motion is used, which consists in periodic trapezoidal back and forth motions realized with the motor actuating bending superposed on a linearly varying mean value (see Figure 4.7). This trajectory includes many changes of directions, in order to exhibit the effects of backlash. The position of the instrument is tracked in the endoscopic image, and the actual trajectory is compared with the modeling obtained with f_{coarse} (pure kinematic model of the instrument (De Donno et al. 2013)), with f_{coarse} combined with backlash estimation only ($f_{coarse}+$ pure backlash) and with the complete estimation ($f_{remain}+$ pure backlash) (see Figure 4.8).

When using f_{coarse} only, the norm of the error in the image shows large pseudo-periodic variations, with a mean distance error of 89 pixels. This is mainly due to the prediction going on either side of the actual position, because the absence of movement at the distal tip at the change of direction is not predicted. The combination of the backlash with f_{coarse} allows to greatly decrease the amplitude of the squares because, at each change of direction, the model predicts the range of motor motion that does not create distal motion. Zoom A in Figure 4.7 shows that the appearance of motion in the image at each cycle is well synchronized between prediction and measurement. However, the prediction shows larger amplitudes of motions than the measurement, and the mean distance error only decreases to 73 pixels. This arises from an incorrect prediction of the rate of variation at the distal side. Namely, the rate is overestimated

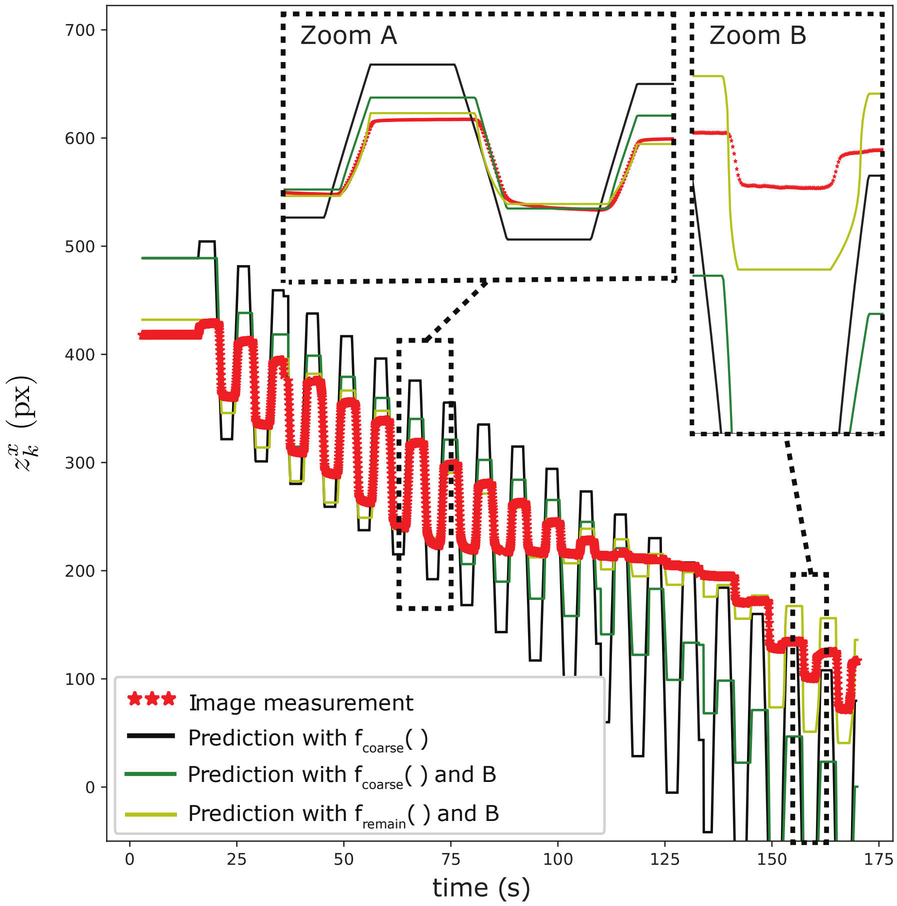


Figure 4.7: Pseudo-periodic trajectory used for testing prediction capability of different models: f_{coarse} (black), $f_{coarse} + B.$ (green) and $f_{remain} + B.$ (yellow). The inlay is a zoom on a back and forth motion, which shows that when backlash is modeled the appearance of motion is well synchronized with the measurement.

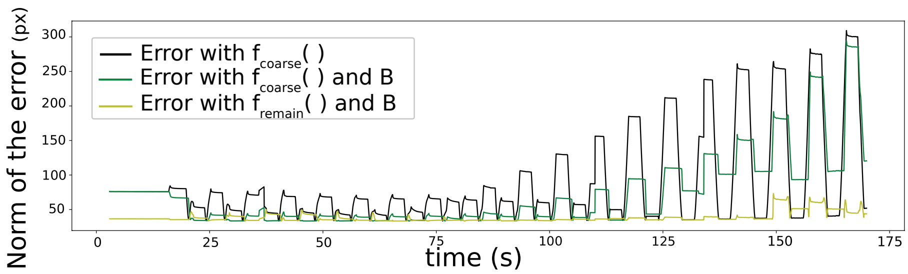


Figure 4.8: Reprojection errors between the measured position of the effector in the endoscopic image and the prediction of effector position provided by f_{coarse} (black), $f_{coarse} + B.$ (green) and $f_{remain} + B.$ (yellow).

because f_{coarse} does not correspond to the actual remaining function. This is especially visible between 150 s and 175 s (see zoom B) because the instrument comes close to the straight configuration, where the actual rate of motion is very low and the discrepancy with f_{coarse} particularly large. Note, however, that the synchronization of motions remains good, indicating that the backlash is well estimated. When using the complete model with f_{remain}, the velocity of distal movement is better predicted, which allows an important reduction of the prediction error to 40 pixels. In zoom B, the remaining function only allows to slightly decrease the error on the image position prediction. The remaining error can be due to the transition movement between backlash and the envelope function or to coupling effects.

It can be noted that the norm of the error never goes under 40 pixels. This comes from errors in the registration between the endoscopic camera and the robot, which mainly affect the vertical position (z^y) of the instrument in the image. In the presented experiments, the motion of the instrument is mainly along the horizontal axis of the image, so that the error on z_k^y is almost constant. The prediction error along the horizontal axis (z_k^x) is therefore significantly reduced from 78 pixels to 17 pixels when using $f_{remain} + B$.

Assessment of compensation capability

The inverse non-linear models have been implemented onto the robotic system. For assessing the effect of compensation, a 2D trajectory (an ellipse) is defined in the task space of the continuum robot. The configuration trajectory (bending, rotation, translation) is computed using the inverse geometric model of the instrument (De Donno et al. 2013). The obtained trajectory is then performed in open-loop and observed by an external camera parallel to the plane of the ellipse. Figure 4.9 provides the qualitative results. The non-linearity compensation allows to largely improve the accuracy of the trajectory.

4.4.4 Discussion

The obtained models can be used to compensate hysteresis and thus facilitate control and improve positioning accuracy (see Figure 4.9). Although the obtained results are satisfactory, it can be observed in figures 4.7, 4.8, and 4.9 that significant errors remain. The sources of these errors are difficult to isolate, but we identified the following items as having the most potential impact on modeling inaccuracies:

- firstly, the estimation of $f_{remain}(.)$ is subject to various modeling errors: camera registration, non-uniform curvature of the robot, and marker position w.r.t. the end-effector to name the main ones;

- secondly, it is assumed that the backlash width is equal to the motor range between the two branches of the hysteretic model (see Figure 4.3, center) such that the system directly switches from the pure backlash to the envelope function of the hysteresis. This is not generally the case since smooth transitions can exist as visible in Figure 4.6b, which leads to underestimated backlash width since motion is detected as early as the first distal motion;

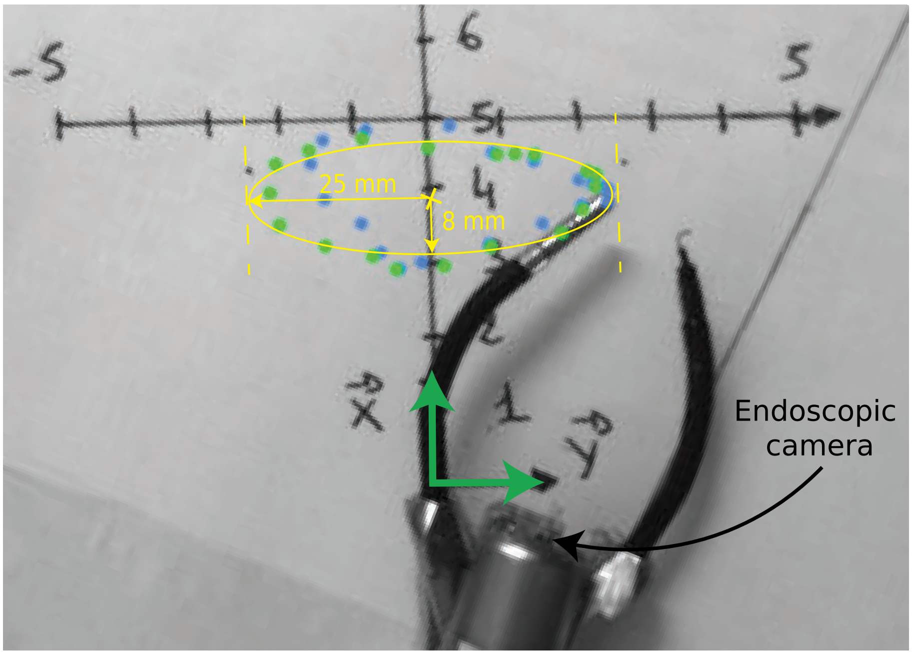


Figure 4.9: Open loop realization of a reference trajectory (yellow), by relying on f_{coarse}^{-1} (blue) and on the inverse model constructed from $f_{remain} + B$. (see Section 4.3.3) (green).

- furthermore, coupling effects between DOF could lead to non-symmetric backlash that cannot be taken into account (i.e., backlash is assumed symmetrical by design);
- finally, the velocity of the actuators have to be limited due to the low acquisition framerate. However, it is possible that other dynamic effects appear at higher velocities, for instance caused by stick-slip phenomena.

It should be noted that most of the aforementioned limitations would also affect other techniques for hysteresis compensation relying on the use of endoscopic images as discussed in the discussion after this chapter.

4.5 Conclusions

In this chapter, we have presented a method for estimating the non-linearities of cable-driven continuum robots, which uses an endoscopic camera conventionally available in endoluminal digestive surgery and which is thus suitable for *in-situ* use with eye-to-hand endoscopic robots. The method is valid for free moving continuum instruments through laboratory experiments, both for predicting the robot behavior and for compensating the non-linearities. One originality of the approach is to separate hysteresis estimation in two parts, and to estimate pure backlash without relying on pose estimation. This is an interesting feature because pose estimation in *in vivo* environments is a complex task (Cabras et al. 2017). Here, backlash estimation is obtained from motor measurements and image measurements only. Tracking a single point on the instrument in the image is sufficient, and there is no need to combine several image information to estimate a pose or a configuration as is realized in (Reilink et al. 2013b; Reilink et al. 2013a; Cabras et al. 2017) or to use data acquired beforehand on the system as proposed in (Baek et al. 2020). Moreover, methods based on models need to handle singularities in the process of 3D information reconstruction. Such singularities can come from the kinematics of the continuum robot (for instance rotation is ill-defined around the straight configuration of the instrument) or from the image Jacobian which links instrument tip motion to its apparent motion in the image.

The method also allows to take into account nonuniform backlash that can appear in some endoscopic tools as shown in figures 2.4 and 4.6b. When identified, pure backlash can be compensated, which can improve user experience during teleoperation. If pose estimation can be obtained, it is possible to estimate the remaining proximal to distal position relation. Interestingly, the proposed approach allows complementing the backlash model, without changing it. Therefore, if pose estimation turns out erroneous because of modeling or measurement errors, the backlash compensation will not be affected. This is an interesting feature compared to other methods where complete hysteresis estimation relies on pose estimation (e.g., Chapter 2 or Reilink et al. 2013b). For the identification of $f_{remain}(\cdot)$, other methods than the one proposed could be used. A recent work by Cursi et al. (2022) used the motion detection approach to identify the backlash (offline from external pose tracking) and to compensate it. They then reacquired motion data with the backlash compensation active and learned $f_{remain}(\cdot)$ using a neural network. Although it is not *in-situ* or online, this work shows that the general approach can be used with different types of sensors and learning methods.

Discussion

Different novel methods for the *in-situ* identification of backlash were proposed. The advantages and drawbacks of each method make them more or less suitable to different scenarios, but the methods we proposed are generic overall. For instance, the FM-DEKF approach was first designed for eye-to-hand endoscopes (i.e., Chapter 2), but extended to the eye-in-hand case in Chapter 3. Similarly, the motion detection approach was developed and tested on eye-to-hand endoscopes in Chapter 4, but could potentially be used in the eye-in-hand case as well: if the endoscope and environment are static, then image motion is sufficient to infer distal robot motion, although it might be quite sensitive to unintentional motions (e.g., physiological).

In the following, we compare the methods we proposed with each other and with other existing methods. Then, we conclude by a discussion about practical issues to consider when using the learned models for backlash compensation.

Comparison of the *in-situ* backlash estimation approaches

As different backlash identification approaches were presented in chapters 2, 3, and 4, it is pertinent to compare them and to highlight their respective strengths and drawbacks. Note that for the comparison of methods, the estimation of $f_{remain}(\cdot)$ in Chapter 4 is not taken into account since it is independent from the motion detection-based approach for backlash width estimation. The main differences between methods for *in-situ* backlash estimation are reported in Table 4.1 and discussed in the following.

Robot modeling and available measurements

A critical difference between the methods, both presented in this thesis and in the literature in general, lies in their dependence on a correct modeling of the robot, including forward kinematics and camera registration. The method based on motion detection presented in Chapter 4 does not require a kinematic model of the robot, contrary to more classical methods based on pose estimation as the one proposed in Chapter 2 or by Reilink et al. (2013b). Model requirements for these methods also include the registration of the endoscopic camera w.r.t. the endoscopic tool, which is not necessarily trivial to obtain (e.g., camera itself mounted on a second flexible endoscopic tool as in Figure 1.6). Therefore, the method we presented in Chapter 4 is advantageous if the kinematic model of the endoscope is unavailable or inaccurate. The same is true with an eye-in-hand camera configuration: a method such as the one proposed by Ott et al. (2011) should be preferred to a model-based method (i.e.,

	Method	General approach	Online	Multi-DOF	Backlash width model complexity	Full hysteresis model	Need for accurate kinematic model	Sensor noise handling
Eye-to-hand	Reilink et al. (2013b)	Model-based learning from pose reconstruction	Yes	Yes	Constant	Yes	Yes	No
Eye-to-hand	Chapter 2	Model-based learning from pose reconstruction	Yes	Yes	**Linear (w.r.t. q)**	Yes	Yes	**Yes**
Eye-to-hand	Chapter 4	Motion detection in image space	No	No	**Generic (non-parametric)**	No (backlash width only)	**No**	No
Eye-in-hand	Ott et al. (2011)	Backlash width measurement in image space	No	No	Constant	No (backlash width only)	No	No
Eye-in-hand	Chapter 3	Model-based learning from SLAM-like approach	**Yes**	**Yes**	**Linear (w.r.t. q)**	No (backlash width only)	Yes	**Yes**

Table 4.1: Main characteristics of the considered image-based *in-situ* backlash model identification methods for eye-to-hand and eye-in-hand flexible endoscopes. The advantages of the methods proposed in chapters 2, 3, and 4 over comparable state-of-the-art methods are highlighted using bold font.

Chapter 3) if the kinematic model is unavailable or very inaccurate. Then, an avenue for future development lies in the kinematic model $\mathcal{K}(\cdot)$ that was assumed correct in chapters 2 and 3, but might in practice be incorrect. If such model inaccuracies can be traced back to estimated model parameters, those could be taken into account by the learning process.

The requirements in terms of measurement availability also differ between backlash identification methods. Those based on pose reconstruction (i.e., Chapter 2 and Reilink et al. 2013b) naturally require image measurements that are sufficiently rich to reconstruct the pose. In the absence of such measurements (e.g., no markers on the endoscopic tool), robot pose estimation-based method cannot be used, but a method such as the one presented in Chapter 4 can still be used since it does not rely on pose reconstruction.

In-situ VS online

Although all methods in Table 4.1 can be used *in-situ* with endoscopic image measurements only, they cannot all be used online. In this context, online means while the robot is performing operational movements, whether automatic or teleoperated. For eye-to-hand robots, the method from Reilink et al. (2013b) and the one proposed in Chapter 2 can be used online as long as there is no contact with the environment, but the method based on motion detection presented in Chapter 4 cannot since it requires specific DOF by DOF motions. The same is true in a eye-in-hand scenario, the method proposed by Ott et al. (2011) cannot be used online due to slow DOF by DOF motion requirements while the method we proposed in Chapter 3 can.

It should be noted that the estimation, prediction, and compensation of backlash are realized for free motions of the robot, but that contacts would impact the validity of all the methods mentioned in this document. Most backlash estimation methods, if used online, are therefore mainly aimed at contactless tasks, such as optical coherence tomography biopsy or laser dissection.

Backlash model complexity

Previous state-of-the-art *in-situ* methods only considered constant backlash width models (Ott et al. 2011; Reilink et al. 2013b), but this is not always sufficient to accurately model the cable actuation of flexible endoscopes. Therefore, increasing the achievable backlash model complexity was one of the main motivations of our work and we proposed in Chapter 2 and 3 approaches to incorporate generic parametric hysteretic models in the *in-situ* learning process. We showed that we could learn backlash width models in the form of a linear function of the actuator position and more complex parametric functions could potentially be used to model the envelope (e.g., parabolic). In Chapter 4, we presented a method based on non-parametric backlash width identification that allows arbitrarily complex backlash width functions. This non-parametric model allows additional flexibility that is sometimes necessary to model systems with especially complex backlash.

Another difference between identification methods lies in the scope of the learned model. The method of Reilink et al. (2013b) and the one presented in chapter 2 both learn a complete hysteresis model that can be used to compensate the backlash and accurately position the robot. On the contrary, the method based on motion detection presented in Chapter 4 only performs the identification of the backlash width and cannot be used to accurately position the robot, but only to compensate the backlash. Although we showed that the backlash width model could be augmented to reconstruct the full model, it requires an independent identification of the remaining function (see Section 4.3).

Also, non-linear phenomena (grossly illustrated in Figure 4.10 and visible in Figure 4.6b) can appear during the transitions from or into the deadzone. These phenomena are not taken into account by any of the *in-situ* identification methods, but they remain a potential issue with real systems as already commented by Ott et al. (2011). These transitions cannot be directly modeled with the motion-detection-based approach presented in Chapter 4 because they would require reconstructing a rate of change of the configuration variable. However, Chatzis et al. (2017b) demonstrated that DEKF filters could be used with Bouc-Wen differential models that explicitly take transitions into account. Then, the FM-DEKF approach could potentially be extended to backlash models based on differential equations (e.g., Bou-Wen hysteresis models) to take such non-linear transitions into account. This is an interesting perspective to build on the work presented in chapters 2 and 3.

Practical considerations for the compensation of backlash

As previously discussed, the presence of non-linear transitions leads to two types of deadzones:

- A pure deadzone entered at the change of motion direction, characterized by a complete lack of distal motion;
- A smooth “non-linear transition” between the pure deadzone and the external envelope. It is characterized by a reduced distal velocity and a varying motion transmission rate (see blue curve in Figure 4.10, left).

However, the backlash identification methods from the literature, as well as those presented in the previous chapters, assume that only a pure backlash is crossed when switching between the envelope branches. The consequences of ignoring the presence of this non-linear transition depends on the identification method. If the method is based on motion detection (i.e., as in Chapter 4), then the identified backlash width is equal to the width of the pure deadzone. Consequently, the identified backlash is narrower than the total backlash defined as the total motor range between the branches of the envelope: the backlash is then underestimated as illustrated in Figure 4.10 (right). On the contrary, if the identification process is based on pose estimation (e.g., Chapter 2 and Reilink et al. 2013b), then the identified backlash width will be close to the real one. In a way, it is the functions underlying the envelope's branches that are identified. Nonetheless, the non-linear effects may introduce a bias such that the backlash width is

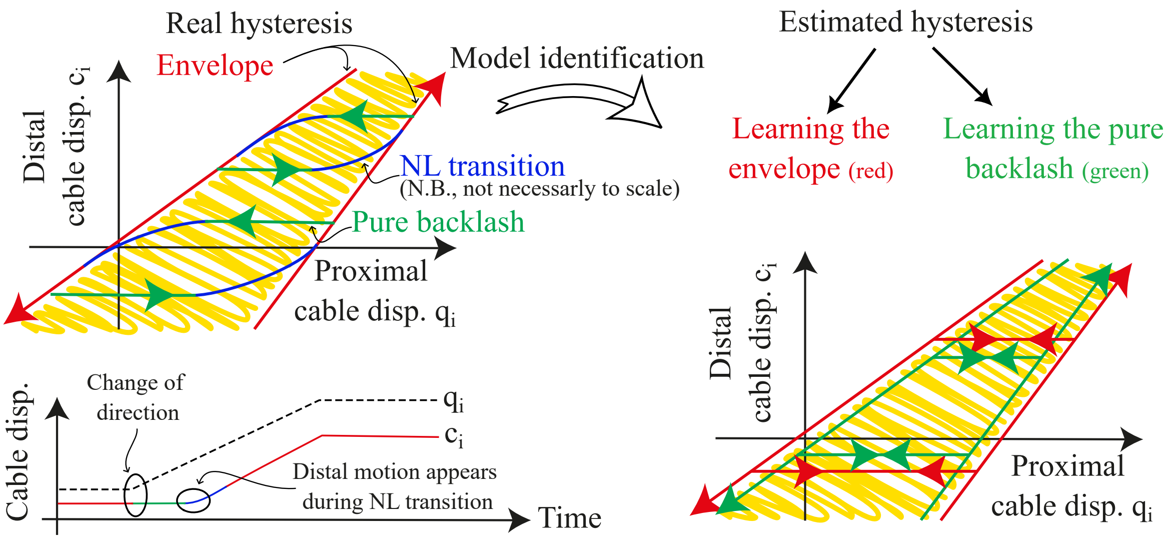


Figure 4.10: Left: illustration of the non-linear smooth transition between the deadzone and the branches of the envelope. Right: the model resulting from the identification process is illustrated, both for the case where pure backlash is identified (e.g., Chapter 4) and the case where the functions of the envelope are identified (e.g., Chapter 2).

slightly underestimated, depending on the range impacted by the non-linear transition (i.e., how large is the non-linear region w.r.t. the pure backlash). Finally, if the backlash is directly extracted from the hysteretic cycle (e.g., graphically), both the pure and total backlash can be extracted as discussed by Ott et al. (2011).

It might then seem that the identification of the total backlash is superior to the identification of the pure backlash, since the model resulting from the former matches the envelope of the real hysteresis better. However, these backlash models are to be used for backlash compensation during operational use of the robot (e.g., teleoperation) and the biases introduced by the non-linear transitions can lead to over-compensation if it is the envelope that is identified. Feedforward compensation approaches consist in inverting the backlash model such that when a change of motion direction is detected, the deadzone is crossed automatically. If it is the pure backlash that is identified, it does not create any problem. However, if the identification process consisted in learning the envelope, the feedforward controller will automatically cross from one branch to another in a short time: consequently, the motor range with the non-linear transition is also crossed, and an undesired distal motion appears. This overcompensation effect was experimentally demonstrated in (Ott et al. 2011) and led the authors to only keep the pure backlash in the model used for compensation. The alternative to cope with this issue is to implement a controller that increases the transmission rate (i.e., velocity gain) when the deadzone is being crossed, without actually crossing it automatically. This approach significantly reduces apparent backlash while avoiding overcompensation (Reilink et al. 2013b). Backlash compensation based on a Bouc Wen model or on a bilinear backlash model (Vaiana et al. 2018) that includes a non-zero transmission rate in the deadzone would also be an option.

Part II

Task and robot model correction from human in-the-loop

Chapter 5

Simultaneous task and robot models learning from user input

Contents

5.1 Related work . . . 74
5.2 Problem formulation . . . 77
5.2.1 Considered setup and modeling . . . 77
5.2.2 User desired trajectory . . . 77
5.2.3 Haptic guidance . . . 78
5.2.4 Learning problem . . . 80
5.3 Proposed approach . . . 80
5.3.1 User in-the-loop as an information source . . . 80
5.3.2 General optimization formulation . . . 81
5.3.3 Recursive implementation using an EKF . . . 82
5.3.4 Practical tuning of hyperparameters . . . 83
5.4 Illustrative simulation results . . . 85
5.4.1 Description of the 2D toy scenario . . . 85
5.4.2 Results with simulated user inputs . . . 86
5.4.3 Discussion . . . 88
5.5 Management of variable time parameters . . . 89
5.5.1 Modeling of the time parameters variability . . . 89
5.5.2 Simulation results . . . 90
5.5.3 Alternative time parameterization . . . 91
5.6 Conclusions . . . 92

In Part I, robot modeling errors have been corrected *in-situ* to improve the robotic assistance. The different methods we proposed were complementary, but they all treated the question of backlash modeling. However, robot modeling errors can take other forms and, in addition, correcting these models *in-situ* would be of interest for medical application. An accurate task model is also critical to many robotic assistive strategies, including haptic or state shared control strategies. Therefore, the learning problem should also be generalized to task modeling. In this chapter, we propose an original approach to perform the online learning of task and robot models. Instead of relying on exteroceptive sensors as in Part I, we propose to exploit information extracted from an operator in-the-loop to perform the online models correction. This is especially relevant to medical telerobotics since a surgeon is typically present to perform the procedure and will be able to correct inaccurate follower robot positioning using visual feedback.

The approach we propose is based on the formulation of the problem as an optimization to minimize the errors between the executed and desired trajectories, both estimated from parametric models. In order to learn the correct parameters of the task registration and of the robot kinematics simultaneously, we model the presence of the user as a constraint relating the task and robot models, and we solve the resulting problem using a Bayesian filter. The problem is formulated in such a way that additional sensor data can also be used for the learning, hence complementing the information extracted from the operator in-the-loop. Furthermore, we use the learned models to provide assistance in the form of haptic guidance, so that the final authority remains the operator.

After providing an overview of the most relevant related works, this chapter introduces the theoretical developments of the proposed method and provides illustrative simulation results. A thorough and comprehensive experimental validation is then provided by the chapters 6 and 7.

5.1 Related work

A vast scientific literature is related to the work presented in this chapter, including early work on compliant trajectory tracking for physical human-robot interaction (pHRI) using impedance control (Hogan 1985)[1]. This type of controller for pHRI can be seen as a way to adapt the robot trajectory from physical interactions with the operator. In the case of impedance control, from an HSC perspective, the reference input of the controller is the reference trajectory and the operator can apply corrective forces if his/her intent differs (Abbink et al. 2012). The stiffness rendered by the controller then relates to the authority of the automation. More involved pHRI strategies have also been proposed to dynamically allocate this authority online, usually through a variable impedance, online from an estimation of model inaccuracies, the operator's skill level, or other relevant metrics (Selvaggio et al. 2021). The allocation of authority can also be tuned offline such that the task model uncertainty is included at the planning/learning stage and then communicated to the operator during HSC. For instance, Zeestraten et al. (2018) learned a trajectory from offline demonstrations and encoded the resulting uncertainty as a covariance that is then used to adapt the stiffness of the haptic guidance during

[1] Impedance control was originally proposed for robot-environment interaction, not necessarily pHRI.

the task execution. Human interventions can also trigger a transition from automatic execution to shared control such that the operator can manually correct the inaccurate execution of a task (Hagenow et al. 2021). Similarly, Losey et al. (2018b) proposed an approach to deform a robot trajectory locally when the operator applies a corrective force. However, none of the works presented so far actually treats the question of task adaptation through learning, but rather the detection and mitigation of physical conflicts between operator and robot.

The question of model learning or correction driven by operator action has been extensively investigated, including in the context of adaptive VF for haptic guidance and, more generally, of pHRI controllers. Although mostly proposed in the context of collaborative robotics where kinesthetic interaction with a human partner is expected, most of the pHRI literature is also applicable to telerobotics, given that the master robot is force enabled, which is typically the case in robot-assisted MIS.

A first strategy is to rely on the operator to explicitly demonstrate the desired corrections. For instance, Restrepo et al. (2017) proposed an approach to update a path used for haptic guidance from demonstrated corrections: the operator physically corrects the path followed by the robot and the task model defined through waypoints is updated accordingly. However, such approaches are more adapted to repetitive tasks, since they require the operator to momentarily stop performing the current task to demonstrate what is the correct task. In (Masone et al. 2014), the authors used the human input to globally deform a previously planned trajectory during its automatic execution, but the operator is then relegated to a supervisory role and does not not control the fine execution. Selvaggio et al. (2018) proposed to adapt the geometry of a VF *in-situ* for robot-assisted surgery from detected tool-tissue interactions, but a distal force sensor is then necessary and most surgical robots have none. Furthermore, it is not applicable to free-air trajectories for which there is no contact to detect.

Previous works also considered online task adaptation strategies based on a library of tasks or even simple primitives. For instance, Raiola et al. (2018) proposed to infer the correct trajectory among multiple ones learned offline from forces applied on the robot by the operator. Another approach consisted in decomposing offline a task in simple primitives (lines) and to choose online the primitive that best matches the operator actions (Aarno et al. 2005). The "active" primitive was then used to provide haptic guidance during teleoperation. Such an online selection among a library of predefined tasks has also been used to provide automatic task completion (Zein et al. 2020). It should be noted that these previous solutions adapt the provided assistance online, but not the underlying models ; it is then not applicable if the correct task is not among those learned offline, for instance due to registration errors. Actual corrections have been predominantely proposed in the context of iterative learning from demonstrations (Kastritsi et al. 2018). Similarly, task adaptation can be achieved by iteratively minimizing the forces applied to the robot by an operator through updates of the reference trajectory tracked by an impedance controller (Xia et al. 2020; Yang et al. 2022).

The works most relevant to us are then those concerned with the online learning of task parameters from the observation of a human partner. Early work focused on the inference of goals in the form of the desired terminal robot position, i.e., a target to reach in Cartesian space (Javdani et al. 2018). The adaptation of actual trajectories was also investigated. Losey et al. (2019) proposed to use corrective forces applied by an operator during the collaborative manipulation of a robot to update a parametric trajectory. This approach is based on a gradient descent that minimizes the interaction force by updating the task parameters. A similar approach was proposed by Bajcsy et al. (2017) to learn online the weights of a parameter-linear reward function from human physical corrections. The adaptive reward function was then used to generate the reference trajectory followed by the robot (Bajcsy et al. 2017; Losey et al. 2018a; Bobu et al. 2020). These works on the online learning of task parameters are the closest to the method we propose in this chapter. However, they are significantly different in the following ways:

- firstly, they are not adapted to refine the potentially inaccurate task registration, which is a critical point in surgical robotics;
- secondly, disagreements about the task execution velocity between operator and automation are not considered. The point-to-point learning approach used in (Bajcsy et al. 2017) or (Losey et al. 2019) assumes that the time parameterization of the trajectory is correct, which is generally not true when the robot is teleoperated by an operator. In a teleoperation context, the haptic guidance would be continuous such that the operator should impose the pace to the automation;
- finally, robot modeling inaccuracies are not considered in previous works about online task correction from interactions with an operator.

It should be noted that this last point is true for every method mentioned so far in this section, the common implicit assumption being that the robot kinematic model is perfectly known or that adequate measurements are available (e.g., possibility to use external localizer). This is valid mostly for industrial-like robots evolving in structured environments, but very much less for surgical robots. In practice, the robot model might also be partially inaccurate, which invalidates the planning of the task expressed in the joint space of the robot and leads to execution errors. Some methods were proposed to adapt the robot model online using teleoperation execution where the state of the robot is usually fully measured (Self et al. 2019; Broad et al. 2020). To the best of our knowledge, simultaneous online learning of task and robot kinematic models has received little attention to date. Existing methods are limited to offline learning. For instance, Pignat et al. (2022) proposed a method to learn simultaneously the correct task and robot kinematic models from human demonstrations, but the learning is then performed offline, not online when the operator actually interacts with the robot to perform the task. The correct models might then change between the time of their offline learning and *in-situ* use.

On the basis of the state-of-the-art presented in this section, the main contributions of the method we are going to develop can be summarized as follows:

- it consists in a unified approach for the online correction of task model parameters, including registration parameters, and robot model parameters from the actions of an operator in-the-loop;

- this approach is well suited to teleoperation with haptic guidance since the task execution velocity is imposed by the operator, not the automation;
- it is possible to include sensor data in addition to the observation of the operator's actions.

5.2 Problem formulation

Before detailing the proposed approach in Section 5.3, we introduce the considered telerobotics scenario along with the different assumptions and models that will be used throughout the chapter.

5.2.1 Considered setup and modeling

From now on, let us consider a robot operated remotely in order to perform a trajectory-following task. Let $x_{m,k}$ be the Cartesian pose of the master robot in its frame of reference $\mathcal{F}_m$ and $x_{s,k}$ be the follower robot Cartesian pose in a world frame $\mathcal{F}_w$ attached to its base. The notation $x_{m,k}$ denotes discrete time such that $x_{m,k} = x_m(t_k)$, with $t_k = kT_s$ and T_s the sampling period. The master pose $x_{m,k}$ is mapped to the follower operational workspace through a linear map $\mathcal{M}(.)$ to generate the follower robot Cartesian reference $x^r_{s,k}$. From this reference, a position controller computes the follower robot joint position reference that is assumed to be ideally tracked by a low-level controller. The forward kinematic model of the follower robot is written as $x_{s,k} = \mathcal{K}(\theta_{r,k}, q_{s,k})$, where $\theta_{r,k}$ is a vector of robot model parameters and $q_{s,k}$ denotes the joint positions. The position controller operates based on an estimate of the parameters $\hat{\theta}_{r,k}$, yielding the kinematic relation described on the block diagram given in Figure 5.1. The estimated robot parameters $\hat{\theta}_{r,k}$ used by the controller can either be fixed (e.g., computed beforehand from a calibration process or given by the robot data sheet) or updated during the execution of the task. This whole process can be seen as a possibly non-linear mapping from $x_{m,k}$ to the follower pose

$$x_{s,k} = \Psi\left(x_{m,k}\right) \tag{5.1}$$

where

$$\begin{aligned} \Psi(x_{m,k}) &= \mathcal{K}\left(\theta_{r,k}, \mathcal{K}^{-1}(\hat{\theta}_{r,k}, x^r_{s,k})\right) \\ &= \mathcal{K}\left(\theta_{r,k}, \mathcal{K}^{-1}(\hat{\theta}_{r,k}, \mathcal{M}(x_{m,k}))\right) \end{aligned} \tag{5.2}$$

5.2.2 User desired trajectory

The user aims at performing a desired task with the follower robot. To do so he/she manipulates the master interface relying on a real-time visual feedback to determine the desired task. This desired follower robot Cartesian trajectory expressed in the world frame is denoted $x^d_{s,k}$. It is assumed to be described by a family of parametric curves g such that

$$x^d_{s,k} = g(\theta_{g,k}, t_k) \tag{5.3}$$

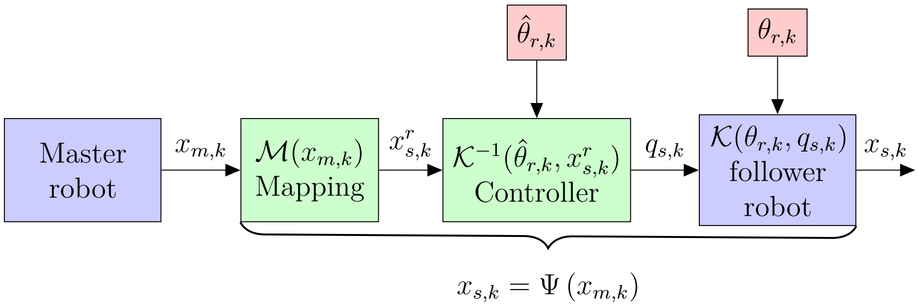


Figure 5.1: Block diagram of the position control of the follower robot. The whole process can be seen as a non-linear mapping $x_{s,k} = \Psi(x_{m,k})$ from master pose $x_{m,k}$ to effective follower robot pose $x_{s,k}$ that depends on both the estimated robot model parameters used by the controller and the real robot model parameters.

where $\theta_{g,k}$ is a set of task parameters. These parameters should typically encode the time parameterization of the task (e.g., the desired velocity) or its registration from a planning frame to the world frame. Any parameterized expression of the form $g(\theta_{g,k}, t_k)$ can be used, as long as it is differentiable w.r.t. time and parameters.

Using the visual feedback, the user will try to adjust the master interface such that the follower robot is at the desired pose at all time. Due to perception biases or intrinsic human limitations (e.g., precision, response time, etc.), there is a time-varying execution error $\epsilon_{h,k}$ between the desired and effective follower robot poses such that

$$x_{s,k} = x^d_{s,k} + \epsilon_{h,k} \tag{5.4}$$

5.2.3 Haptic guidance

The haptic guidance essentially helps the user to perform the desired trajectory by applying guidance forces or torques to the master haptic interface. These forces are computed to render a mechanical impedance whose equilibrium position is the guidance reference pose $x^g_{m,k}$, itself evaluated from a model of the operator's desired trajectory. Note that the desired trajectory is not known and has to be estimated. As the desired trajectory is modeled by a family of parametric curves, estimating the desired trajectory amounts to estimating the parameters $\theta_{g,k}$. Therefore, the estimated desired pose in the world frame is

$$\hat{x}^d_{s,k} = g(\hat{\theta}_{g,k}, t_k) \tag{5.5}$$

where $\hat{\theta}_{g,k}$ is the current estimate of $\theta_{g,k}$. Since the guidance forces are applied to the master robot, the guidance reference must be mapped back into the master workspace such that

$$x^g_{m,k} = \mathcal{M}^{-1}\left(\hat{x}^d_{s,k}\right) \tag{5.6}$$

The complete teleoperation setup is represented in Figure 5.2 along with the desired trajectory $x^d_{s,k}$ and the guidance reference $x^g_{m,k}$ resulting from Equation (5.6).

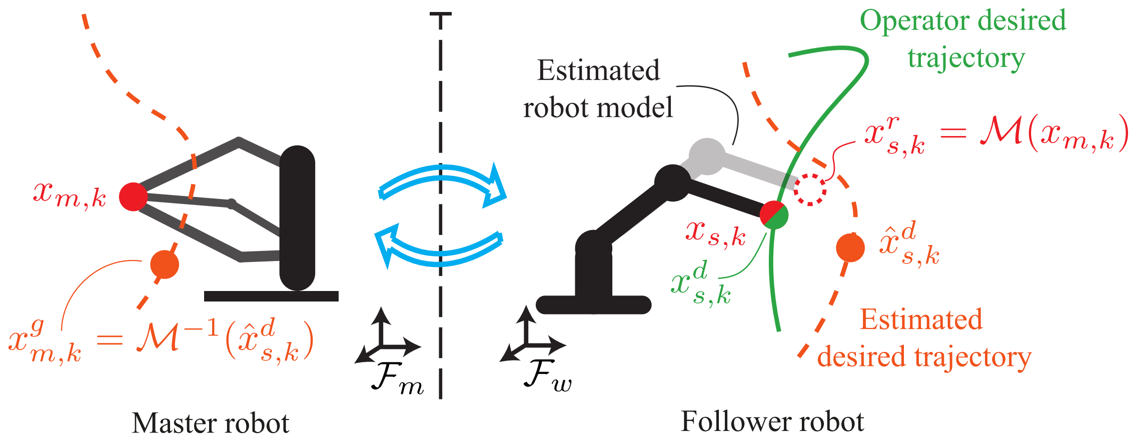


Figure 5.2: Teleoperation setup where the operator remotely controls the follower robot to execute a desired trajectory (green). An estimation of this desired trajectory (orange) is used to provide a master side haptic guidance, though this estimation is incorrect. The resulting guidance error is compounded by follower robot positioning errors resulting from erroneous kinematic modeling (gray vs. black).

An accurate estimation of the parameters $\theta_{r,k}$ and $\theta_{g,k}$ is critical to the performance of the guidance. It is easy to understand that an inaccurate task parameters estimate would cause the guidance to be inaccurate. The effect of errors on the kinematic parameters is probably more difficult to understand, but it is actually similar. Equation (5.1) reduces to $\Psi(x_{m,k}) = \mathcal{M}(x_{m,k})$ *if and only if* $\hat{\theta}_{r,k} = \theta_{r,k}$ as in this case Equation (5.1) is written as $x_{s,k} = \mathcal{K}\left(\theta_{r,k}, \mathcal{K}^{-1}(\theta_{r,k}, \mathcal{M}(x_{m,k}))\right)$. The user will be guided towards $x^g_{m,k}$ under the assumption that the model of the robot used by the controller is accurate as defined in Equation (5.6). Therefore, even assuming that the follower robot pose desired by the user $x^d_{s,k}$ is perfectly known, an error on the robot model parameters leads to inaccurate guidance as, from equations (5.1) and (5.6) it then comes that:

$$\begin{cases} \hat{\theta}_{r,k} \neq \theta_{r,k} \\ \hat{\theta}_{g,k} = \theta_{g,k} \\ x_{m,k} = x^g_{m,k} \end{cases} \implies \quad x_{s,k} = \Psi\left(\mathcal{M}^{-1}\left(x^d_{s,k}\right)\right) \neq x^d_{s,k}$$

In that case, the guidance is also experienced as inaccurate by the operator who will have to compensate for the follower robot's positioning error. Additionally, if the kinematic model is strongly non-linear or/and the robot modeling errors are large, the trajectory that the user will have to follow at the master side will be deformed and potentially non-intuitive as illustrated in Figure 5.3.

Regardless of the source of the guidance inaccuracies, the user will experience forces pulling towards undesired directions, which is an inconvenience and can even lead to degraded task achievement. Either way, the haptic guidance stops being assistive when the guidance errors increase. It is therefore critical to correct the task and kinematic parameters when either or both are incorrect and to do so online because the optimal parameter values can change during the task execution (e.g., because the task

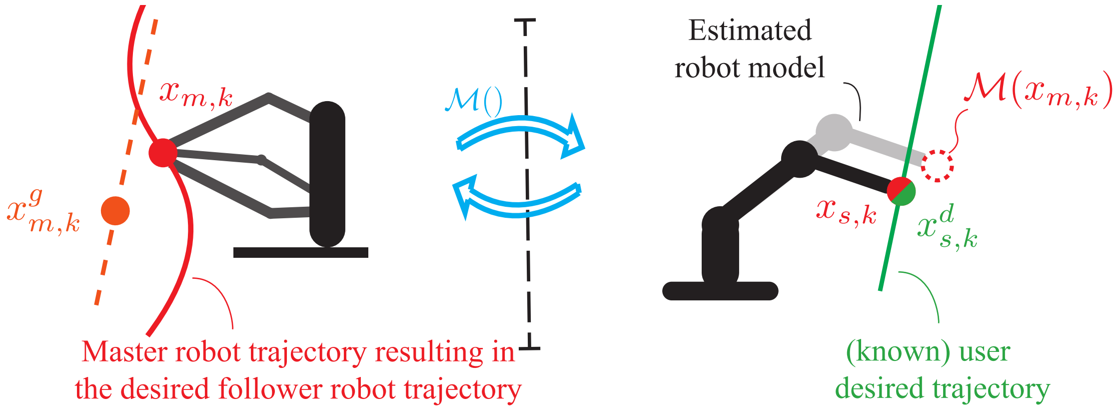


Figure 5.3: Example of how a robot positioning error can lead to a deformation of the desired master robot trajectory. In the absence of such errors, the desired master pose is simply the mapping of the desired follower robot pose into the master workspace $\mathcal{M}^{-1}(x^d_{s,k})$.

registration changes). In the following, an approach to online parameters learning is presented to cope with task and robot model inaccuracies.

5.2.4 Learning problem

The learning problem can be stated as two simultaneous parameters estimations that, in their simplest forms, would be written as:

$$\theta_r^* = \arg\min_{\theta_r} \sum_{i=0}^{k} ||x_{s,i} - \mathcal{K}(\theta_r, q_{s,i})||^2 \tag{5.7}$$

$$\theta_g^* = \arg\min_{\theta_g} \sum_{i=0}^{k} ||x^d_{s,i} - g(\theta_g, t_i)||^2 \tag{5.8}$$

assuming that the parameters do not change over time for a considered time horizon $[0; t_k]$. Note that neither the follower robot's Cartesian pose nor the desired one might be directly measured. Usually, only partial observations are available, if any. Possible sensor information can come from camera image measurements, force sensing, etc., but this information can sometimes be poor or intermittent. It is then advantageous to exploit user inputs because they are dependent on both the real task and robot kinematic parameters.

5.3 Proposed approach

5.3.1 User in-the-loop as an information source

The user will manipulate the master robot in such a way that the follower robot's pose $x_{s,k}$ is as close as possible to the desired position $x^d_{s,k}$. Hence, $x_{s,k}$ can be seen as a noisy observation of $x^d_{s,k}$ as defined in Equation (5.4). However, if the follower robot Cartesian pose is not measured by an external sensor, the only observation available for learning is the joint position $q_{s,k}$. Then, we propose to use the presence of the operator to formulate an optimization problem that minimizes the discrepancy between the estimation of the executed Cartesian trajectory and the trajectory predicted by the task model.

5.3.2 General optimization formulation

The general learning problem is then stated as the minimization of a loss function that penalizes the violation of the constraint resulting from Equation (5.4). The aforementioned constraint is then written as

$$\begin{aligned} &\mathcal{K}(\theta_{r,k}, q_{s,k}) - g(\theta_{g,k}, t_k) = \epsilon_{h,k} \\ &\text{with } \epsilon_{h,k} \sim \mathcal{N}(0, \Sigma_{h,k}) \end{aligned} \tag{5.9}$$

where $\mathcal{N}(.)$ represents a normal distribution such that $\Sigma_{h,k}$ is the covariance of the (presumably) normally distributed zero-mean error $\epsilon_{h,k}$. Additional observations about the state of the robot and the environment can be incorporated in the form of additional terms in the cost function. Let z_k be a vector containing the observations acquired at time t_k that depend on the parameters and can be predicted as

$$z_k = h(\theta_k) + w_k, \;\; w_k \sim \mathcal{N}(0, R_k) \tag{5.10}$$

where $h(.)$ is the observation model, $\theta_k = [\theta_{g,k}^T \; \theta_{r,k}^T]^T$ contains the different model parameters, and w_k is the measurement error of covariance R_k. Small parameters variations arising from online changes in the environment, in the kinematics of the robot, or in the preferences of the human operator can be modeled by a Gaussian noise v_k with covariance Q_k such that

$$\theta_{k+1} = \theta_k + v_k, \;\; v_k \sim \mathcal{N}(0, Q_k) \tag{5.11}$$

The online learning problem can be seen as an optimization performed at time step k considering the measurements $z_{0:k-1}$ and $q_{s,0:k-1}$ acquired at discrete times $t_{0:k-1}$, i.e., the sequence of times $\{t_0;\, t_1, \cdots, t_{k-1}\}$. The optimization is performed w.r.t. the sequence of parameters $\theta_{0:k}$. If all available observations are considered, it can be written as:

$$\theta^*_{0:k} = \arg\min_{\theta_{0:k}} l_\theta(\theta_0) + \sum_{i=0}^{k-1} l_i(\theta_i, \theta_{i+1}, z_i) \tag{5.12}$$

with

$$\begin{aligned} l_\theta(\theta_0) &= \|\hat{\theta}_0 - \theta_0\|^2_{P_0^{-1}} \\ l_i(\theta_i, \theta_{i+1}, z_i) &= \|\mathcal{K}(\theta_{r,i}, q_{s,i}) - g(\theta_{g,i}, t_i)\|^2_{\Sigma_{h,i}^{-1}} + \|\theta_{i+1} - \theta_i\|^2_{Q_i^{-1}} + \|z_i - h(\theta_k)\|^2_{R_i^{-1}} \end{aligned}$$

where the function $l_\theta(\cdot)$ penalizes initial parameters update, with $\hat{\theta}_0$ the initial guess for the parameters value at $k = 0$ and P_0 the covariance of this initial estimate. The term

$l_\theta(\cdot)$ allows to include prior knowledge about the initial parameters distribution. Its effect fades over time as more observations are available. The function $l_i(\cdot)$ is a weighted sum of prediction errors computed according to the models defined by equations (5.9), (5.10), and (5.11). The learning framework can be used in conjunction with a vast class of non-linear state/parameters estimation methods such as moving horizon estimation (Rawlings 2015), or other non-linear Bayesian estimation techniques (e.g., extended or unscented Kalman filters, particle filters, etc.). This flexible approach allows to use the presence of an operator to augment the information available for learning. Although not required, sensor measurements can be included for learning and, due to the Bayesian formulation, the fusion of observations coming from multiple sensors is streamlined, with the sole requirement that the covariance of the measurements can be estimated.

5.3.3 Recursive implementation using an EKF

In order to demonstrate how the learning method can be implemented, an EKF is devised. It is one of the simplest ways to solve the optimization problem. The cost defined by Equation (5.12) is computed based on the current estimates and measurements, so the filter only stores the current parameters estimated values and covariance, computed under approximations of Gaussian noise and linearized models. More precisely, a variant of the EKF is used, which includes an exponential forgetting of past observations. The state transition and observation model are defined as

$$\theta_{k+1} = \theta_k + v_k,\ \ v_k \sim \mathcal{N}(0, Q_k) \tag{5.13}$$

$$\bar{z}_k = \bar{h}(\theta_k) + \bar{w}_k,\ \ \bar{w}_k \sim \mathcal{N}(0, \bar{R}_k) \tag{5.14}$$

where $\bar{z}_k$ is the observation vector z_k augmented with the *pseudo-measurement* associated to the constraint (5.4). This concept of *pseudo-measurements* has been used in the past to include equality constraints in state estimation schemes to improve tracking by explicitly including kinematic or dynamic constraints (De Geeter et al. 1997; Simon 2010). The derivative of this constraint modeled by the *pseudo-measurement* is also used in order to exploit the joint velocity measurements $\dot{q}_{s,k}$ in addition to the joint positions $q_{s,k}$ such that

$$\bar{z}_k = \begin{bmatrix} \epsilon_{h,k} \\ \dot{\epsilon}_{h,k} \\ z_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ z_k \end{bmatrix} \tag{5.15}$$

$$\bar{h}(\theta_k) = \begin{bmatrix} \mathcal{K}(\theta_{r,k}, q_{s,k}) - g(\theta_{g,k}, t_k) \\ J_{\mathcal{K}}(\theta_{r,k}, q_{s,k})\dot{q}_{s,k} - \dot{g}(\theta_{g,k}, t_k) \\ h(\theta_k) \end{bmatrix} \tag{5.16}$$

$$J_{\mathcal{K}}(\theta_{r,k}, q_{s,k}) = \left.\frac{\partial \mathcal{K}(\theta_{r,k}, q_s)}{\partial q_s}\right|_{q_s = q_{s,k}} \tag{5.17}$$

and

$$\bar{R}_k = \begin{bmatrix} \Sigma_{h,k} & 0 & 0 \\ 0 & \dfrac{2}{T_s^2}\Sigma_{h,k} & 0 \\ 0 & 0 & R_k \end{bmatrix} \tag{5.18}$$

where the term $\frac{2}{T_s^2}\Sigma_{h,k}$ is the noise covariance matrix of $\dot{\epsilon}_{h,k}$. This expression arises from a naive time derivative of the noise model defined in Equation (5.9) that assumes a constant $\Sigma_{h,k}$, but in practice other expressions can be chosen. The parameters values and covariance estimates are computed with the EKF from this augmented observation using the update rule

$$\hat{\theta}_{k|k-1} = \hat{\theta}_{k-1} \tag{5.19}$$

$$\hat{P}_{k|k-1} = (1+\alpha_k)\hat{P}_{k-1} + Q_k \tag{5.20}$$

$$\hat{\theta}_k = \hat{\theta}_{k|k-1} + K_k\left(\bar{z}_k - \bar{h}(\hat{\theta}_{k|k-1})\right) \tag{5.21}$$

$$\hat{P}_k = \hat{P}_{k|k-1} - K_k\bar{H}_k\hat{P}_{k|k-1} \tag{5.22}$$

where the subscript $k|k-1$ denotes a prior estimate at sample time k, $\alpha_k \geq 0$ is a factor tuning the weight given to past observations, K_k is the Kalman gain matrix such that

$$K_k = \hat{P}_k\bar{H}_k^T\left(\bar{H}_k\hat{P}_k\bar{H}_k^T + \bar{R}_k\right)^{-1} \tag{5.23}$$

and $\bar{H}_k$ is the Jacobian of the augmented observation model w.r.t. the parameters :

$$\begin{aligned} \bar{H}_k &= \frac{\partial}{\partial\theta}\bar{h}(\theta)\bigg|_{\theta=\hat{\theta}_k, t=t_k} \qquad (5.24)\\ &= \begin{bmatrix} \dfrac{\partial}{\partial\theta_g}\bar{h}(\theta) & \dfrac{\partial}{\partial\theta_r}\bar{h}(\theta)^T \end{bmatrix}_{\theta=\hat{\theta}_k, t=t_k} \\ &= \begin{bmatrix} -\dfrac{\partial}{\partial\theta_g}g(\theta_g,t) & \dfrac{\partial}{\partial\theta_r}\mathcal{K}(\theta_r, q_{s,k}) \\ -\dfrac{\partial}{\partial\theta_g}\dot{g}(\theta_g,t) & \dfrac{\partial}{\partial\theta_r}J_{\mathcal{K}}(\theta_{r,k}, q_{s,k})\dot{q}_{s,k} \\ \dfrac{\partial}{\partial\theta_g}h(\theta) & \dfrac{\partial}{\partial\theta_r}h(\theta) \end{bmatrix}_{\theta=\hat{\theta}_k, t=t_k} \end{aligned}$$

This version of the EKF is often referred to as a fading memory Kalman filter (Simon 2010) and the so-called fading factor α_k in Equation (5.20) allows to discard past observations. When $\alpha_k = 0$, the method is equivalent to a classical EKF. As the value of α_k increases, less weight is given to past observations.

5.3.4 Practical tuning of hyperparameters

Hyperparameters have to be tuned in order to reach satisfactory learning performances. Some of them may differ depending on the implementation of the proposed method, such as the fading factor that is specific to the fading memory EKF. But overall, the reasoning would be the same with other implementations. Although hyperparameters would have to be tuned according to the situation, some general insights are provided thereafter to illustrate their effect on the learning performance. Experimental determination of optimal hyperparameter values for a given scenario would allow for a better tracking of the parameters, but a compromise can be found such that the learning performs well on a wide variety of tasks.

The fading factor α_k allows to tune the weight given to past observations such that their effect becomes negligible after a certain time. This allows to cope with potential parameter drift and modeling errors arising from the linearization of the possibly non-linear models (Simon 2010). Large values of α_k improve the learning performance, but reduce the robustness to execution errors. This is explained by the fact that if all past data is discarded, a local deviation from the desired trajectory (i.e., execution errors) will cause the estimated parameters $\hat{\theta}_k$ to degrade or even oscillate. The value of α_k should therefore be chosen as a trade-off between robustness and learning rate. Interestingly, the fading factor α_k can be tuned independently from the chosen models parameterization.

An intuitive way to characterize the exponential fading memory effect over time is to consider the relative weight given to past observations. The decay time constant τ of this weight is

$$\tau = \frac{T_s}{\ln\left(1+\alpha\right)} \tag{5.25}$$

Equivalently, Equation (5.25) can be rearranged as $\alpha = e^{T_s/\tau} - 1$ and using this expression, a value of α can be computed from the desired decay time constant τ (Nelson 2000).

If there is no particular hypothesis on the state process noise covariance Q_k, it can simply be assumed that the parameters are fixed, such that the process noise covariance Q_k is null. The fading memory implementation will ensure that past observations are gradually discarded. If a specific structure of Q_k is known, an ad-hoc process noise matrix can be used. Note that if there is no fading, i.e., $\alpha_k = 0$, then the process noise should be introduced through Q_k or the learning will eventually stop as the estimated parameter covariance P_k decreases.

The human execution covariance Σ_h should be chosen to model the human behavior and reflect the expected execution errors considering the sampling period. Generally, a larger covariance matrix slows down the learning by decreasing the weight given to the demonstrations. Overestimating the execution errors allows to smooth the parameters estimation to obtain a more gradual learning. Similarly, the initial estimated parameter covariance P_0 should ideally encode the real uncertainty over parameter estimates to reach an optimal performance, but most often this information is unknown. Then, the initial covariance should be underestimated to avoid sudden updates of the parameters at startup when the information is the least reliable. The transient behavior at startup only depends on the chosen values for P_0 and Σ_h, and the proposed implementation with the fading memory limits the influence of inaccurate covariance modeling over time.

5.4 Illustrative simulation results

In this section, both the learning problem and the proposed approach to solve it are illustrated through a simulation. The simulated scenario is very simple, but sufficient to demonstrate how the proposed method can be used to correct task and robot models simultaneously using the presence of an operator.

5.4.1 Description of the 2D toy scenario

A planar serial robot with 2 DOF is simulated. Its end-effector position $x_{s,k}$ is computed using the forward kinematic model given the joint positions $q_{s,k}$ and the length L_1 and L_2 of the two links of this serial robotic arm. Only L_1 is known and the robot kinematic model $\mathcal{K}(\theta_{r,k}, q_{s,k})$ is parameterized such that $\theta_{r,k} = \theta_r = L_2$ is a constant robot parameters. Note that the location of the robot tip depends on the robot joint values and on the robot model parameters such that $x_{s,k} = \mathcal{K}(\theta_r, q_{s,k})$. If some parameters are not accurate (i.e., L_2 in this scenario) the estimated robot position $\hat{x}_{s,k} = \mathcal{K}(\hat{\theta}_{r,k}, q_{s,k})$ differs from the real one. A 2D Cartesian trajectory is planned in a frame of reference $\mathcal{F}_t$ (see Figure 5.4a) and subsequently registered in the world frame $\mathcal{F}_w$ attached to the base of the robot (see Figure 5.4b). The task consists in following this reference while it is not properly registered in $\mathcal{F}_w$. The ground truth time-dependent desired trajectory of constant parameters θ_g is of the form

$$\begin{aligned} x^d_{s,k} &= g(\theta_g, t_k) \\ &= \begin{bmatrix} t_x \\ t_y \end{bmatrix} + \begin{bmatrix} \cos r_z & -\sin r_z \\ \sin r_z & \cos r_z \end{bmatrix} \Gamma(a + bt_k) \end{aligned} \tag{5.26}$$

where $\theta_g = \begin{bmatrix} a, \, b, \, t_x, \, t_y, \, r_z \end{bmatrix}^T \in \mathbb{R}^5$ is a vector of parameters, $\Gamma(\psi) \in \mathbb{R}^2$ is a smooth curve parameterized by its arc length $\psi \in [0; \psi_{\max}]$. The parameters a and b encode the time parameterization of the task under the assumption that the execution velocity b is constant.

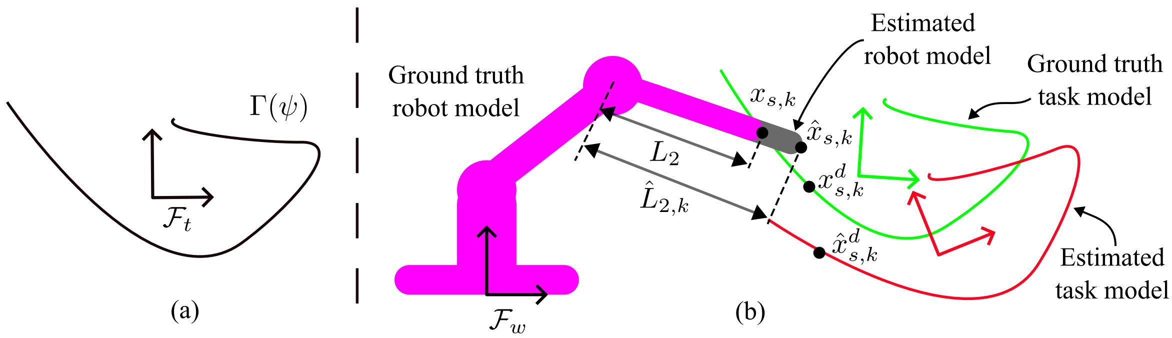


Figure 5.4: Scenario with simulated planar task and robot. (a) The path $\Gamma(\psi)$ is defined in a frame of reference $\mathcal{F}_t$. (b) The path is subsequently registered in the world frame $\mathcal{F}_w$ attached to the base of the robot before its execution.

A scenario where there is a consequent error on the initial estimated parameters, but where the correct parameters do not change over time is considered. It is assumed that a human operator is teleoperating the robot to execute the planned trajectory such that $x_{s,k} = x^d_{s,k} + \epsilon_{h,k}$. A perfect operator is simulated such that $\epsilon_{h,k}$ is null. The articular positions $q_{s,k}$ are then computed such that $q_{s,k} = \mathcal{K}^{-1}(\theta_r, x_{s,k})$. It should be noted that the simulation is not at any realistic scale (see Figure 5.5), it uses normalized unitless distances. It aims at showing that it is possible to extract information from user inputs when there are known task execution constraints such as robot kinematics and a preoperative plan that is followed by an operator.

In the following simulations, the filter is implemented with a sampling time period $T_s = 0.025$ s (i.e., 40 Hz). The decay time constant of the fading memory mechanism is set to $\tau = 20$ s such that α_k is constant and equal to 0.0013 and the process noise covariance matrix Q_k is set to $Q_k = 0_{6\times6}$. This means that after 45 seconds, the weight given to an observation has decreased by 95%. The covariance matrix $\Sigma_{h,k}$, modeling the execution errors introduced by the simulated operator, is set to $\Sigma_{h,k} = \sigma_h^2 I$, where $\sigma_h = 0.01$ with normalized units. Finally, the covariance matrix P_o is arbitrarily chosen as a diagonal matrix modeling initially independent estimated parameters of small standard deviations except for the initial position on the path encoded with $\hat{a}_0$ that is larger to account for the initial uncertainty such that

$$\mathrm{diag}(P_0) = \begin{bmatrix} \underbrace{0.01^2}_{\text{on } \hat{a}_0} & 10^{-9} & 10^{-9} & 10^{-9} & 10^{-9} & 10^{-9} \end{bmatrix} \tag{5.27}$$

5.4.2 Results with simulated user inputs

As visible in Figure 5.5, the simulated robot (pink) follows the ground truth desired trajectory (green) that is the correctly registered path $\Gamma(\psi)$ parameterized with the correct time parameters. The estimation of this desired trajectory (red) is significantly inaccurate, with parameter estimation errors of about 10% (see Figure 5.6). Finally, the initial estimation of the robot model parameter L_2 is also off by roughly 10%.

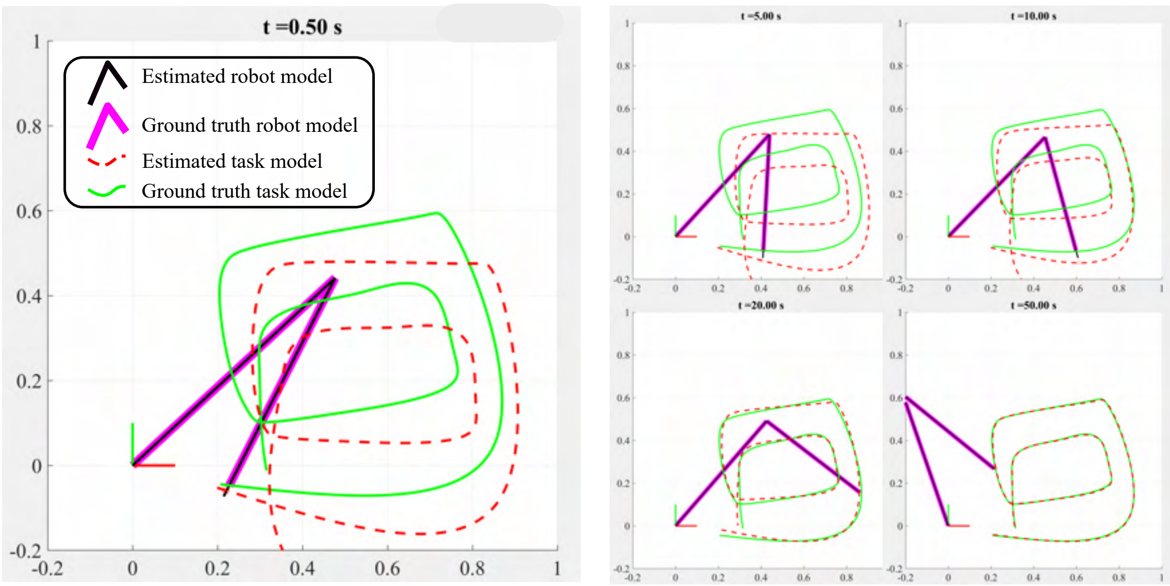


Figure 5.5: Evolution of the task and robot models at different $t_k \in \{0.5; 5; 10; 20; 50\}s$. See Video 5.1 for full animation.

As the estimated execution error $\hat{x}_{s,k} - \hat{x}^d_{s,k}$ (see Figure 5.8) is minimized by the learning algorithm, the task and robot parameter estimation errors are reduced over time (see Figure 5.6). This includes the errors on time parameters a_k and b_k estimation (see Figure 5.7). The estimated parameter $\hat{b}_k$ that encodes the desired execution velocity quickly converges to the correct value, but the estimated time offset $\hat{a}_k$ only converges towards the correct value later. The delayed convergence of $\hat{a}_k$ is explained by the fact that learning the time parameterization amounts to finding on the underlying path the closest point to the estimated robot position $\hat{x}_{s,k}$, which cannot be done until said path and robot position are correct. This happens after one minute of simulation, when the task and robot model parameters have converged towards their correct values (see Figure 5.7), such that the task and robot position prediction errors, respectively $\|x^d_{s,k} - \hat{x}^d_{s,k}\|$ and $\|x_{s,k} - \hat{x}_{s,k}\|$ in Figure 5.8, become null.

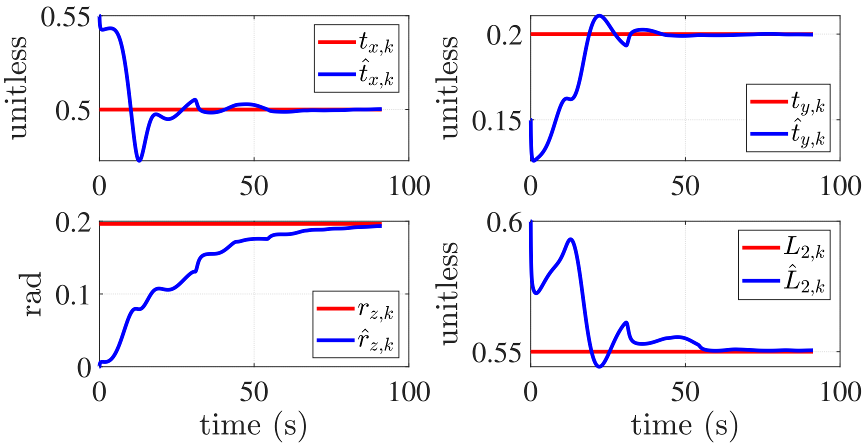


Figure 5.6: Parameter estimation over time.

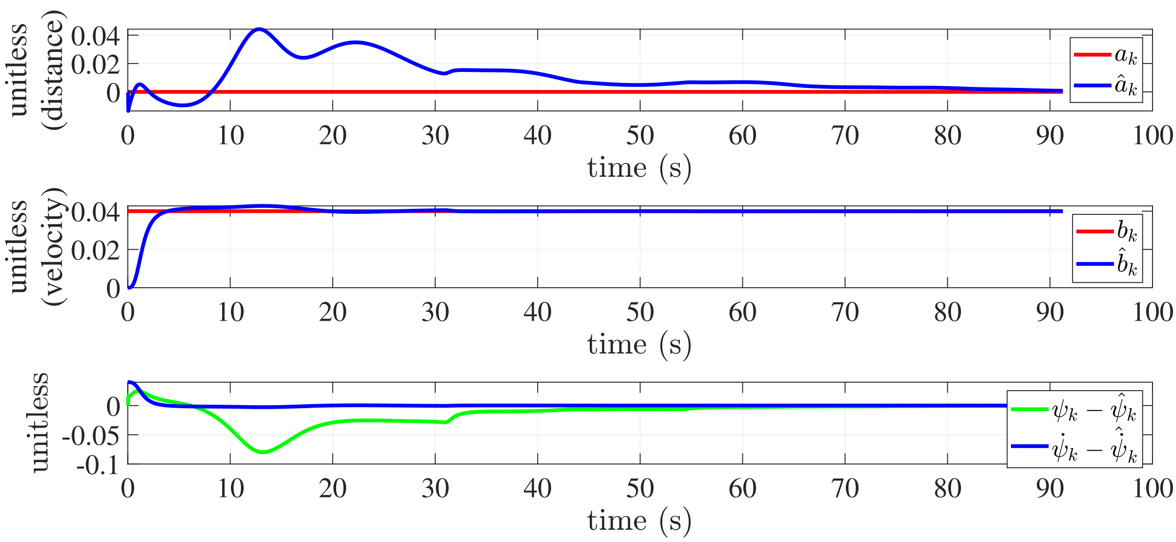


Figure 5.7: Time parameters estimation and resulting estimation error on $\psi_k = a_k + b_k t_k$ and $\dot{\psi}_k = b_k$.

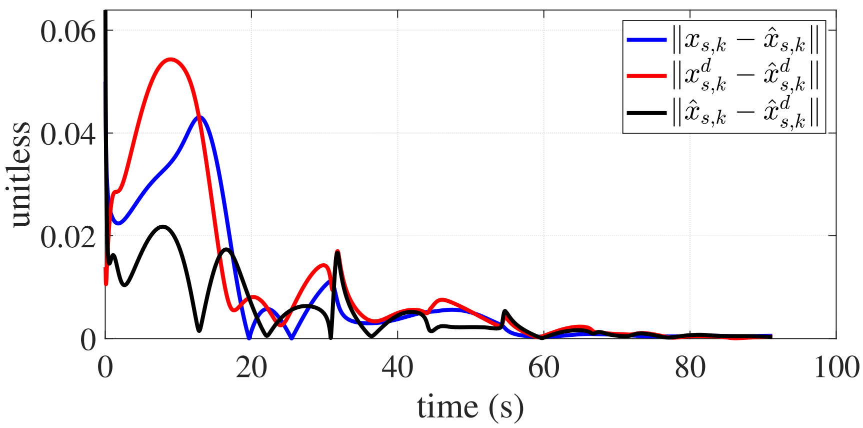


Figure 5.8: Evolution of learning metrics.

5.4.3 Discussion

The simulation illustrates the fact that the proposed approach can successfully extract information from the presence of an operator in-the-loop to correct online the task registration and robot modeling inaccuracies. Furthermore, the desired execution velocity is also learned such that the pace is not imposed by the automation.

One can notice in figures 5.6 and 5.7 that some estimated parameters vary quickly at startup, with a near-instantaneous variation of $\hat{a}_k$, $\hat{b}_k$, $\hat{t}_{x,k}$, and $\hat{L}_{2,k}$ over the span of about 1 s. In the case of the time parameter estmates $\hat{a}_k$ and $\hat{b}_k$, it is the intended behavior. The initial position along the path is incorrect and it is therefore quickly corrected (i.e., $\hat{a}_k$ update). Similarly, the initial execution velocity estimate is incorrect ($\hat{b}_0 = 0$) and is updated quickly. The larger initial uncertainty on $\hat{a}_0$ introduced in P_0 (see Equation 5.27) facilitates a fast time parameters update, explicitly for $\hat{a}_k$ and indirectly for $\hat{b}_k$ that is correlated to $\hat{a}_k$.

However, the initial variation of $\hat{t}_{x,k}$ and $\hat{L}_{2,k}$ is due to the fact that the simulation starts with the robot already at the desired position, hence introducing a consequent initial prediction error. In a scenario with a real operator, the robot would more realistically start at $\hat{x}_{s,0} = \hat{x}^d_{s,0}$ and the correction of the task by the operator would not be instantaneous, hence mitigating this effect.

The other limitation of this simulation is the constant execution velocity, which is highly unlikely in practice, and the unmodeled errors that a real operator would introduce though the learning method itself does suppose that this errors exist. It should be noted that simply introducing a Gaussian noise on $\epsilon_{h,k}$ is of little to no interest since human errors would not be Gaussian. However, modeling accurately the behavior of the operator is out of scope here. The method will instead be evaluated directly with real human operators in the next chapter. However, the question of the possibly variable execution velocity has to be treated for our method to be applicable to real-world scenarios.

5.5 Management of variable time parameters

In practice, the time registration of the task can be inaccurate due to the user slowing down, delaying before starting the gesture, or pausing during the task execution. The estimation error on the time parameters resulting from these events may propagate to all other parameters. As an illustration of the need for a reactive tracking of the time parameters, the execution velocity $\dot{\psi}_k = b_k$ is set to zero for $t_k \in [50; 55]$ to simulate the worst-case scenario when the operator suddenly stops. As shown in Figure 5.9, this results in a abrupt, but temporary change of the correct values for the time parameters a_k and b_k. The filter cannot cope with this sudden change and the estimated time parameters become very inaccurate. As a result, this estimation error propagates to the other parameters, for instance $\hat{t}_{x,k}$ (see Figure 5.9). The full animation of the evolution of the task and robot models is provided as complementary material (see Video 5.2).

A naive solution would consist in decreasing the decay time constant of the forgetting mechanism (i.e., increase the value of α), but this would decrease the adaptation time of all parameters equally. In this case, not only a sudden change of the correct time parameters would introduce errors on the other parameters, but the filter would also be less robust overall since less data would be used for learning. Therefore, it is necessary to model the possible variations of the time parameters independently from the tuning of the global filter convergence rate.

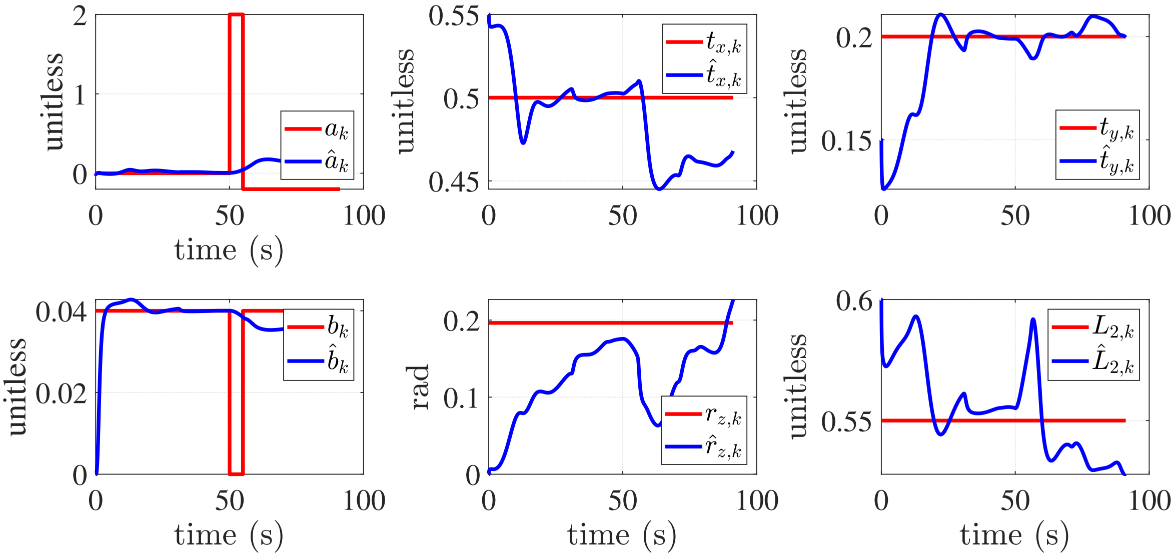


Figure 5.9: Simulated pause during the task execution at $t = 50$s. The time parameters (i.e., a and b) estimation becomes inaccurate as the filter does not adapt quickly enough, which introduces significant errors on the other estimated parameters.

5.5.1 Modeling of the time parameters variability

Tracking the correct time parameters amounts to track the current position along the path $\psi_k = a_k + b_k t_k$ and the current velocity $\dot{\psi}_k = b_k$. In order to improve this tracking, we propose to model the possible variation of $\dot{\psi}_k$ with the process noise covariance matrix Q_k of the filter such that

$$Q_k = \begin{bmatrix} Q_{(a,b),k} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.28}$$

These additional terms only impact the process noise covariance associated to the time parameters a_k and b_k. The matrix $Q_{(a,b),k}$ has to be carefully chosen, since simply modeling additive Gaussian noise on the parameter a_k would ignore the correlation between the two time parameters

The transition from the current position and velocity along the path to the time parameters is

$$\begin{bmatrix} a_k \\ b_k \end{bmatrix} = \begin{bmatrix} 1 & -t_k \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \psi_k \\ \dot{\psi}_k \end{bmatrix}. \tag{5.29}$$

Therefore, the modeled process noise covariance can be propagated from $\dot{\psi}_k$ to the time parameters such that

$$Q_{(a,b),k} = \begin{bmatrix} 1 & -t_k \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \sigma_{\dot{\psi}}^2 \end{bmatrix} \begin{bmatrix} 1 & -t_k \\ 0 & 1 \end{bmatrix}^T = \begin{bmatrix} t^2\sigma_{\dot{\psi}}^2 & -t\sigma_{\dot{\psi}}^2 \\ -t\sigma_{\dot{\psi}}^2 & \sigma_{\dot{\psi}}^2 \end{bmatrix} \tag{5.30}$$

where $\sigma_{\dot{\psi}}$ is chosen considering the shortest time the user would need to stop. The acceptable values range from 0, equivalent to a hypothesis that the pace is near-constant, to the value of the velocity measurement standard deviation, which would model the worst-case theoretical scenario: the user can stop in only one time period.

5.5.2 Simulation results

The improved time parameters tracking scheme is implemented with $\sigma_{\dot{\psi}} = 10^{-4}$ in the same conditions as previously: a sudden pause is simulated at $t_k = 50$ s and lasts for 5 s. In Figure 5.10, one can observe that the time parameters a_k and b_k are correctly tracked and quickly updated when the pause occurs. The convergence time of b_k is similar to that of the first simulation (see Figure 5.6, bottom), which is to be expected since b_k still cannot converge towards its correct value until the other task and robot parameters have also converged.

The main difference between figures 5.9 and 5.10 is that in Figure 5.10, when the pause occurs, the prediction errors are correctly identified as being a change in the time parameters. As a result, the remaining parameters are not impacted, as evidenced by the near-constant value of $\hat{t}_{x,k}$ and $\hat{r}_{z,k}$ during the few seconds when the execution is paused. The full animation of the evolution of the task and robot models is provided as complementary material (see Video 5.3).

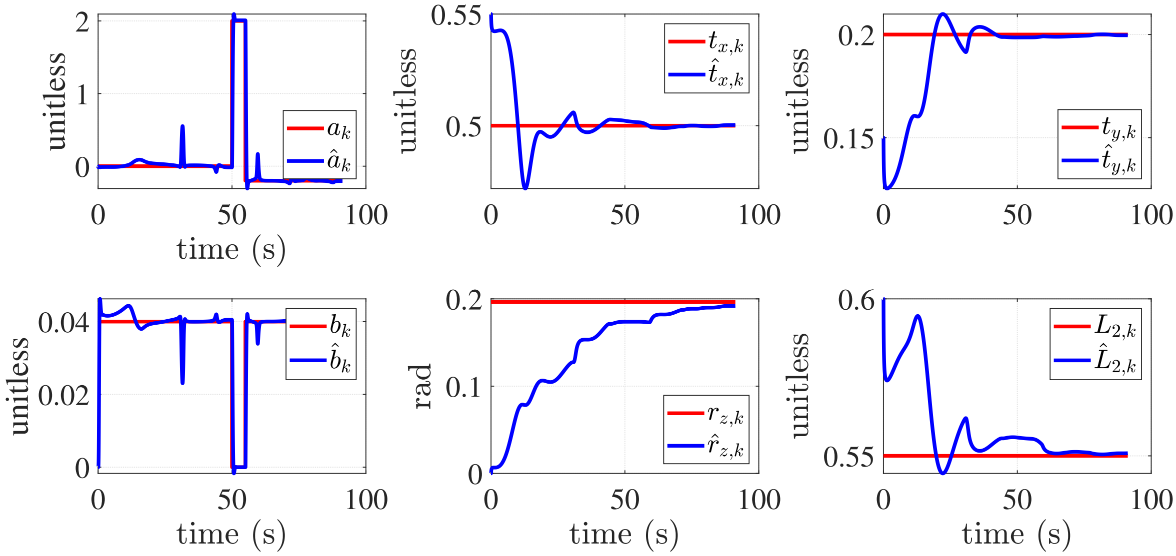


Figure 5.10: Simulated pause during the task execution at $t_k = 50$s. Process noise is injected to cope with variable execution velocity. The time parameters are correctly tracked by the filter and, as a result, no error is introduced on the other parameters estimation when the task execution is paused.

5.5.3 Alternative time parameterization

It should be noted that there is an alternative way to parameterize time evolution. The desired trajectory $x_s^d(\theta_g, t_k)$ could have been directly modeled using a time-variable state variable ψ_k, such that, at each time step, $\psi_{k+1} = \psi_k + T_s b_k$ where b_k is the parameter encoding the desired execution velocity. Of course, the state transition model would then have to be modified to perform the integration of b_k, but this is a minor change that would only affect the EKF filter in Equation (5.19). The two parameterizations are equivalent, but the choice made in this work aims at only using stationary parameters, hence removing the need for a state transition model and facilitating extrapolation when the model is later used for haptic guidance. Additionally, this facilitates the implementation of practical features such as a pause in the task execution or synchronization. However, it has two drawbacks compared with the alternative modeling with ψ_k included as a time-variable state variable. Firstly, it leads to a more complex covariance matrix resulting in Equation (5.30). When modeling ψ_k as a state variable, there is supposedly no correlation between ψ_k and b_k such that a single term σ_ψ^2 in the matrix Q_k is sufficient to model the time parameters uncertainty. Furthermore, explicitly using t_k in the matrix Q_k might pose numerical problems for very long tasks, as t_k will get very large. However, the considered tasks are short and this problem is not relevant in the scenarios considered for experimental validation. We ascertained through simulations and experiments that both modeling choices are equivalent, and we will then use $\psi_k = a_k + b_k t_k$ in the following.

5.6 Conclusions

We proposed an approach for the simultaneous correction of task and robot models. Simulation results show that the correct model parameters can indeed be retrieved from the operator actions and partial knowledge of the task being performed. We also proposed an extension to cope with variable execution velocity, which is a necessary feature if the approach is to be used with a real operator.

The simulation results demonstrate that the improved learning method can cope with variable execution pace imposed by the operator. Even in the worst case for which the operator suddenly stops, the filter correctly tracks the time parameters. The convergence time of these parameters can be tuned by changing the value of $\sigma_{\dot{\psi}}$, independently from the overall convergence rate of the filter. Beyond improving learning, this feature is also interesting for haptic guidance. Large values of $\sigma_{\dot{\psi}}$ will make the time-parameters tracking very reactive such that the operator is guided towards the closest point on the path underlying the trajectory $\hat{x}^d_{s,k}$. Therefore, the guidance would behave like a virtual fixture. For smaller values of $\sigma_{\dot{\psi}}$, the dynamics of the filter will smooth the evolution of $\hat{\psi}_k$, which would tend to also smooth the motion of the operator.

These results illustrate that using the proposed method, both the task and robot models parameters can be adjusted from the user actions only. However, we only tested the learning method with simulated data. The following chapter will further evaluate the performance of the proposed method with actual teleoperation data, while addressing theoretical and practical challenges.

Chapter 6

Use-case: Online task and hand-tool registration during haptic guidance

Contents

6.1 Description of the considered scenario 94
6.2 Identifiability of the task and robot parameters 97
6.3 Experimental validation of the learning method 98
6.3.1 Learning performance with a human in the loop 98
6.3.2 Influence of the fading factor 103
6.3.3 Robustness to parameters shift 104
6.4 Comparison with non-adaptive guidance 105
6.4.1 Experimental protocol 105
6.4.2 Results 106
6.4.3 Analysis 109
6.4.4 Necessity for joint robot and task models corrections 109
6.5 Conclusions 110

In the previous chapter, a method for the simultaneous learning of task and robot model parameters was presented, and the main features of the algorithm were demonstrated with a simulated two-links planar robot. In the following, the method is further evaluated in a telerobotics scenario with a general-purpose 7 DOF robot teleoperated by a user. We chose a generic path-following task that consists in remotely drawing a path using visual feedback. The long term application of our work remains robot-assisted MIS, but the proposed experimental setup is more suited to objective task performance assessment. Nonetheless, it should be noted that the setup presented in this chapter could also be implemented with a surgical robot, although with many more practical issues.

During the realization of the task by the operator, a haptic guidance is provided on the basis of a model of the path to be drawn. However, the task registration, as well as the registration of the tool w.r.t. the end effector it is attached to, are inaccurate. The learning method proposed in Chapter 5 is then used to improve an initially incorrect haptic guidance. The aim of this experimental validation is two-fold: to validate the learning method in a practical scenario and evaluate the effectiveness of the adaptative haptic guidance scheme.

6.1 Description of the considered scenario

We implemented the position teleoperation using a Franka Emika Panda robot teleoperated by a Force Dimension Omega 7 haptic interface. Visual feedback was provided by a camera (webcam C930e, Logitech). The path to be drawn was printed on a tilted (30 deg.) planar support as depicted in Figure 6.1. In this way, the position of the support in the horizontal plane of the world frame and its orientation around the vertical axis affect the task along all three dimensions.

The follower robot end effector was equipped with a force/torque sensor (nano 43, ATI), a pen-holder, and a pen (see Figure 6.1.B). An impedance controller tracks the pen-holder Cartesian pose reference imposed by the operator at the master station (redundancy and inverse kinematics are handled by the controller). The end-effector orientation is constant and computed such that the pen remains normal to the drawing plane. The remaining 3 DOF that correspond to the position of the pen tip are controlled to perform the drawing task. Since the orientation of the follower robot is fixed, the master interface is used as a 3 DOF device and its orientation is ignored. A controlled constant force of 1 N is applied to the pen to guarantee permanent contact with the surface, which leaves 2 DOF for the operator to control. The force control normal to the surface limits the cognitive load that more complex force feedback (even admittance/impedance control) could generate in the presence of haptic guidance. It also prevents possible interferences between force feedback and haptic guidance that can sometimes compensate one another (Smisek et al. 2015). Although the operator can only control 2 DOF of the follower robot, the trajectory to be performed on the master side, as well as the provided haptic guidance, are three-dimensional. The end-effector pose of the robot $x_{ee,k}$ corresponds to the position of a point attached to the pen-holder, i.e., to the base of the pen. Besides, $x_{s,k}$ is the position of the tip of the pen actually used to perform the drawing task.

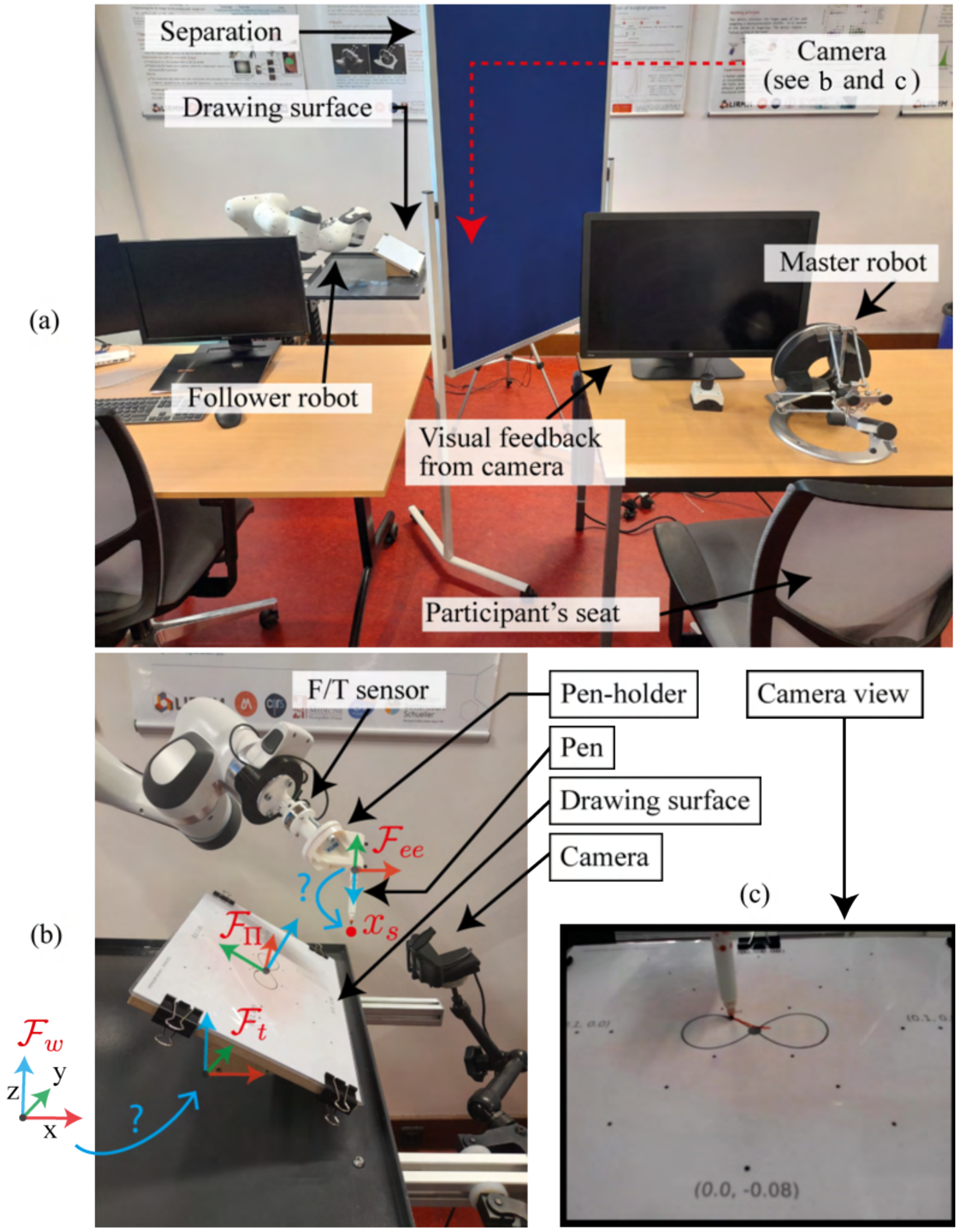


Figure 6.1: (a) Experimental setup. (b) Close view of the task support, the robot end effector, and the pen. The main frames of reference used in the following sections are represented. $\mathcal{F}_\Pi$ is an additional frame aligned with the drawing plane used for convenience when displaying results. The frame attached to the tip of the pen is omitted for clarity. (c) Visual feedback from camera available to the human operator to perform the teleoperation task.

In the proposed experimental setup, a tool registration model defines the relationship between the position of the end effector $x_{ee,k}$ and that of the tip of the tool $x_{s,k}$, such that the transformation between $x_{ee,k}$ and $x_{s,k}$ is a translation along the Z axis of the end-effector frame. The robot model parameter $\theta_{r,k} = L_k$ is the distance between the pen holder position and the tip of the pen. The tool registration model is combined with the known forward kinematic model to obtain the follower robot model $x_{s,k} = \mathcal{K}(\theta_{r,k}, q_{s,k})$. This scenario is similar to what could happen in another context if a gripper was used to hold a tool: the exact tool configuration w.r.t. the end effector could indeed be only partially known.

Let $\Gamma(\psi) \in \mathbb{R}^3$ be a smooth path planned in a planning frame $\mathcal{F}_t$ and subsequently registered in the world frame $\mathcal{F}_w$. In the experiment, the rigid transform from $\mathcal{F}_w$ to $\mathcal{F}_t$ results from the combination of translations along X and Y axes, respectively denoted as $t_{x,k}$ and $t_{y,k}$, and a rotation about Z axis with angle $r_{z,k}$ as in the previous chapter (see Figure 6.1.b). The time parameterization of the task is modeled by an affine function of time, such that $\psi(t_k) = a_k + b_k\, t_k$ as was presented in Chapter 5. The task model is then:

$$x^d_{s,k} = g(\theta_{g,k}, t_k) = \begin{bmatrix} t_{x,k} \\ t_{y,k} \\ 0 \end{bmatrix} + {}^wR_t(r_{z,k})\Gamma(a_k + b_k t_k) \tag{6.1}$$

where $\theta_{g,k} = \begin{bmatrix} a_k, \, b_k, \, r_{z,k}, \, t_{x,k}, \, t_{y,k} \end{bmatrix}^T$ is the set of task parameters and ${}^wR_t(r_z) \in SO^3$ is the rotation matrix between frames $\mathcal{F}_w$ and $\mathcal{F}_t$. The task parameters to be refined are therefore the position and orientation of the support in $\mathcal{F}_w$ and the two time parameters (i.e. 5 parameters in total).

The ROS2 framework was used to handle the different multi-rate components (see Figure 6.2). Haptic guidance was computed according to Section 5.2.3 and implemented using the Force Dimension SDK. The stiffness of the haptic guidance is set to 300 N.m^{-1} and the damping is chosen so that the equivalent system is a damped mass-spring system with a damping coefficient of 0.9. The learning process takes as inputs both the current execution time and the follower robot measurements with sampling period $T_s = 0.02$ s and sends the current task and robot parameter estimates to the high-level controller as detailed in Figure 6.2. This high-level controller computes the guidance reference and the follower robot reference from, respectively, the estimated task model and the estimated tool registration model. It also manages the position-position mapping between master and follower. This architecture is implemented on a Linux platform (i7 / 32GB ram).

In the experiments, the fading factor is set to a constant value $\alpha_k = \alpha = 10^{-3}$ such that the associated decay time constant is $\tau = 20$s. It is a good choice if the parameters do not vary a lot, because the errors introduced by the human operator are filtered out over a span of nearly one minute. Results illustrating the effect of α are reported in Section 6.3.2. The covariance matrix of the execution error introduced by the operator is set to $\Sigma_{h,k} = \sigma_h^2 I$, where $\sigma_h = 5$ mm and P_o is arbitrarily chosen as a diagonal matrix modeling initially independent estimated parameters of standard deviations equal to 10 mm for $\hat{a}_0$, 10 mm.s^{-1} for $\hat{b}_0$, 1 deg. for $\hat{r}_{z,0}$, and 1 mm for $\hat{t}_{x,0}$, $\hat{t}_{y,0}$ and $\hat{L}_0$.

The model $\psi(t_k)$ is learned online to accommodate for the velocity and delays imposed by the operator. Nevertheless the model may become inaccurate when the

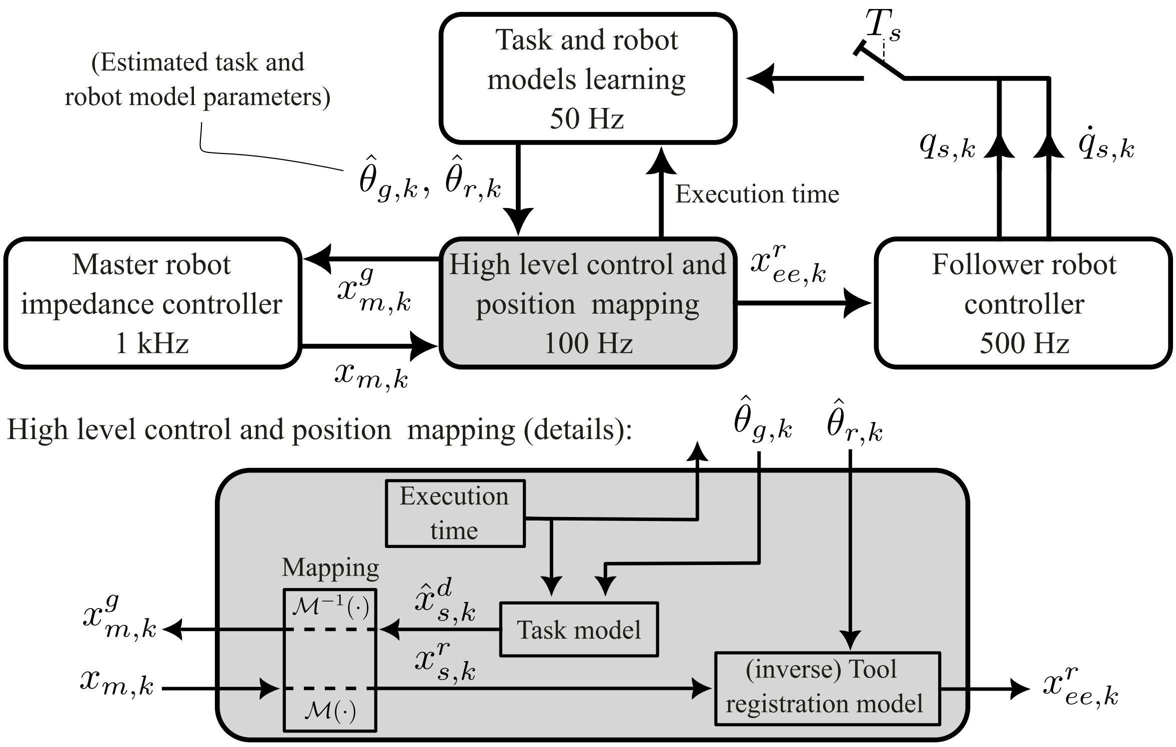


Figure 6.2: Overview of the software architecture and main variables. Details are provided for the high-level controller handling the mapping from $\mathcal{F}_m$ to $\mathcal{F}_w$, the generation of the guidance reference $x^g_{m,k}$, and the generation of the follower robot end-effector Cartesian reference $x^r_{ee,k}$.

operator is slowing down, delaying before starting the gesture, or pausing during the task execution. In order to ensure that the tracking of $\psi(t_k)$ is done in a reactive way, the possible change in pace is modeled with the process noise variance of $\dot{\psi}$ set to $\sigma_{\dot{\psi}} = 0.4$ mm.s^{-1}. As detailed in Section 5.5.1, this results in additional terms in the process noise covariance matrix Q_k that only impact the process noise covariance associated to the time parameters a_k and b_k.

6.2 Identifiability of the task and robot parameters

The task and robot models parameters have to be identifiable to converge towards the true values. Identifiability is a notion close to observability, which is used to analyze a system in order to find if its state can be inferred from the observation of the input and output. The parameters structural identifiability can be verified before the task execution under the assumption that the operator will perform the correct trajectory. To do so, we write the learning problem as a dynamic system with $q_{s,k}$ as the input and $\epsilon_{h,k}$ as the measurement (i.e., the operator is modeled as a sensor); the developments are detailed in Appendix C. Then, we build the observability matrix using so-called extended Lie derivatives (Karlsson et al. 2012) and analyze its rank. We used the *STRIKE GOLDD toolbox* (Villaverde et al. 2019) to compute the observability matrix and its rank symbolically. For this analysis, the correct task and robot models parameters are assumed constant.

We found the parameters to always be (locally) structurally identifiable except when the desired task execution velocity is null or when the desired task is a straight line (i.e. no curvature). The former is explained by the fact that if $\dot{\psi} = 0$, then the operator is not providing any information. The later is explained by the equivalent effect of the time parameter a and the translation of the path underlying $x^d_{s,k}$ when the path is a line. In this case, an execution error in the direction tangent to the trajectory $x^d_{s,k}$ can be equally explained by an estimation error on a_k or by a combination of errors on t_x, t_y, and L. Consequently, when $\Gamma(\psi)$ is a line, only the orientation of the task r_z and desired velocity $\dot{\psi} = b$ are identifiable. This means that with an arbitrary task model as implemented for the experiments, the task and robot parameters are potentially intermittently unidentifiable when the local curvature is beyond a certain range (see Appendix C). But this local lack of observability does not impact the global performance of the learning as long as it indeed stays intermittent, because the EKF filters information over a time horizon. In the present case, this time window can be tuned by the fading factor α. Therefore, if the fading factor is chosen correctly, only two scenarios should lead to a failure during the parameters learning:

1. the human operator does not move. However, this scenario could be detected and solutions to cope with it can be implemented (e.g., pause the learning if the execution velocity is below a given threshold);

2. the task model is such that the parameters are unidentifiable most of the time (i.e., the path is a straight line).

The second scenario is more challenging, as even if it can be predicted a priori, it would be difficult to decide at which point the model parameters are not “identifiable enough.” However, it might be possible to monitor the estimated covariance matrix and implement a fault-detection process that would stop the learning or reset the filter.

It is worth mentioning that the observability analysis we performed is structural, meaning that it does not take into account the learning implementation and the measurement noise. In a real-world application (e.g., in the following sections), the observations will be noisy and, in our case, potentially biased due to the presence of a human operator in-the-loop. However, the information would also be richer since the task will be more complex than a simple arc, hence improving the observability.

6.3 Experimental validation of the learning method

6.3.1 Learning performance with a human in the loop

The performance of the learning method is demonstrated on a task consisting in drawing a treble key (see Figure 6.4). An operator manipulates the haptic interface to follow the path printed on the surface with the pen attached to the follower robot. The estimated task and robot models, and consequently the generated haptic guidance, are initially incorrect. Three arbitrary initial task registration errors are considered.

In order to assess how the learning method would perform in the most favorable scenario, the learning performance is also assessed for a perfect execution of the task at a constant pace, which is not achievable perfectly by a human operator. To do so, the end-effector reference $x^r_{ee,k}$ sent to the robot controller is computed using the correct models, estimated from offline registration and tool calibration. This condition is referred to as **AUTO**, whereas the proposed method with an operator is referred to as **AG** (Adaptive Guidance).

The ground truth path is given by the template printed treble key. The desired trajectory $x^d_{s,k}$ used as a ground truth is computed offline once the task execution is over such that, at all time, $x^d_{s,k}$ is the closest point from $x_{s,k}$ on the ground truth path (see Figure 6.3). This computation is necessary to account for different execution paces because, while the spatial registration can be calibrated beforehand, the correct time registration is the one imposed by the human operator during the task execution. The execution error is defined at each time step as the deviation from the correct path and computed as $||\epsilon_{h,k}||$ where $||\cdot||$ denotes the Euclidean norm. The task and robot model prediction errors at each time step are computed as $||\tilde{x}^d_{s,k}||$ and $||\tilde{x}_{s,k}||$ respectively, with $\tilde{x}_{s,k} = x_{s,k} - \hat{x}_{s,k}$. These metrics depend on the ground truth registration of $\mathcal{F}_t$, which is obtained through an offline calibration procedure consisting in manually pointing two dozen points on the drawing surface with the robot.

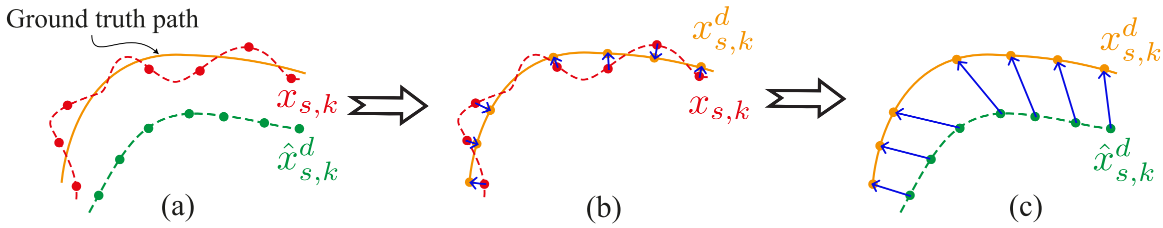


Figure 6.3: (a) Illustration of the collected trajectories $x_{s,k}$ and $\hat{x}^d_{s,k}$. (b) The desired trajectory $x^d_{s,k}$ is computed by finding on the ground truth path the points closest to $x_{s,k}$ (for each sample), which then results in errors $||\epsilon_{h,k}||$ (see b) and $||x^d_{s,k} - \hat{x}^d_{s,k}||$ (see c).

In Figure 6.4, the paths underlying the desired trajectory $x^d_{s,k}$ and its estimation $\hat{x}^d_{s,k}$ acquired during an **AG** scenario are displayed in $\mathcal{F}_\Pi$ for different times t_k. At startup, $\hat{x}^d_{s,k}$ is significantly different from $x^d_{s,k}$ due to the parameter estimation errors and the operator is therefore initially guided along an incorrect trajectory. The drawing plane being at an angle w.r.t. the horizontal plane in $\mathcal{F}_w$, the task prediction error is three-dimensional. Therefore, the incorrect estimation of the orientation r_z impacts not only the in-plane orientation of the estimated desired trajectory, but also the predicted desired position along the normal to the plane. This can be observed in Figure 6.4 where the position and orientation of the initial estimated task model projected in the XZ plane are incorrect. Regardless of guidance errors, the operator can perform the desired task, although execution errors corresponding to imperfect task realization are introduced. For the three **AG** executions (i.e., for the three initial registration errors), this error can be as large as 7 mm at startup. It can reach 2.5 mm locally afterwards, and it is 0.62 ± 0.09 mm on average (see Figure 6.5). As the operator performs the

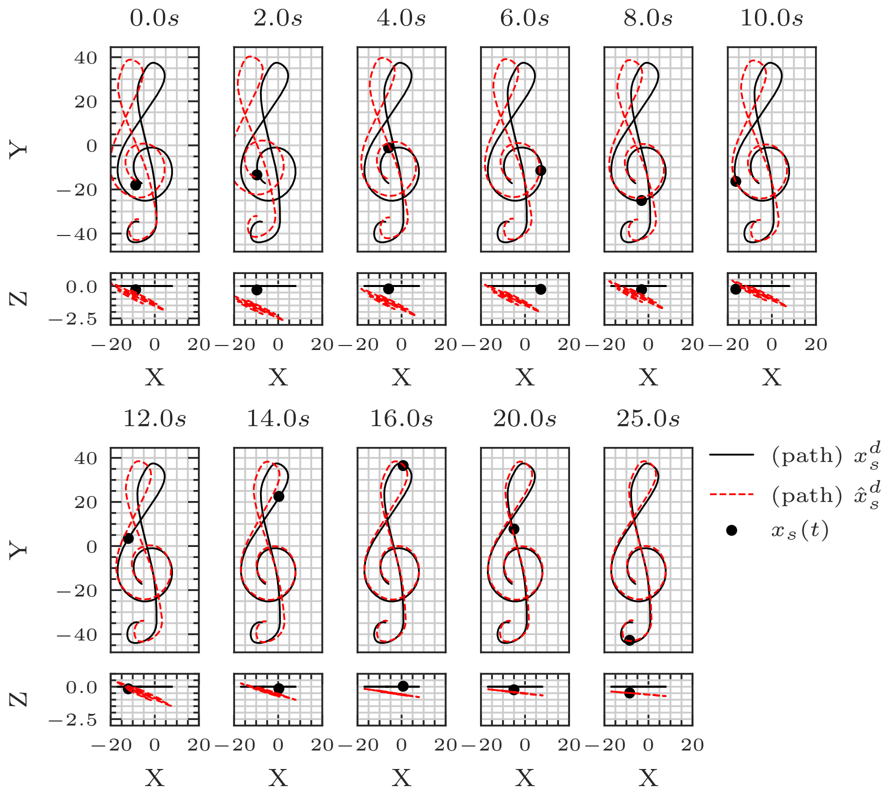


Figure 6.4: Estimated task model evolution over time for the scenario **AG** with initial parameters estimation errors of $\tilde{\theta}_{g,0} = [\, \tilde{a}_0, \tilde{b}_0, \tilde{r}_{z,0}, \tilde{t}_{x,0}, \tilde{t}_{y,0} \,]^T = [\, *, \, *, \, 10, \, -2, \, -5 \,]^T$ in $[\, *, \, *, \, \text{deg.}, \, \text{mm}, \, \text{mm} \,]$ and $\tilde{\theta}_{r,0} = \tilde{L}_0 = 10$ mm. The task model is represented as the path underlying $\hat{x}^d_{s,k}$ and all units for X, Y, and Z are in mm.

task, the models are refined by the learning process and the estimated trajectory $\hat{x}^d_{s,k}$ gradually converges towards $x^d_{s,k}$, even if the optimal parameters are not yet reached when the task ends (see Figure 6.6). The task prediction error (see Figure 6.5) decreases over time, which can also be observed in Figure 6.4 where the task model converges towards the correct ones. The estimated robot position $\hat{x}_{s,k}$, initially incorrect by 11 mm along the Z axis of $\mathcal{F}_\Pi$ also converges towards its ground truth value as the estimation of L is updated (see Figure 6.6). Note that, in the present experiment, L is the only initially unknown robot model parameter and it encodes the length of the pen such that the robot model estimation error is simply $||x_{s,k} - \hat{x}_{s,k}|| = \tilde{L}_k$.

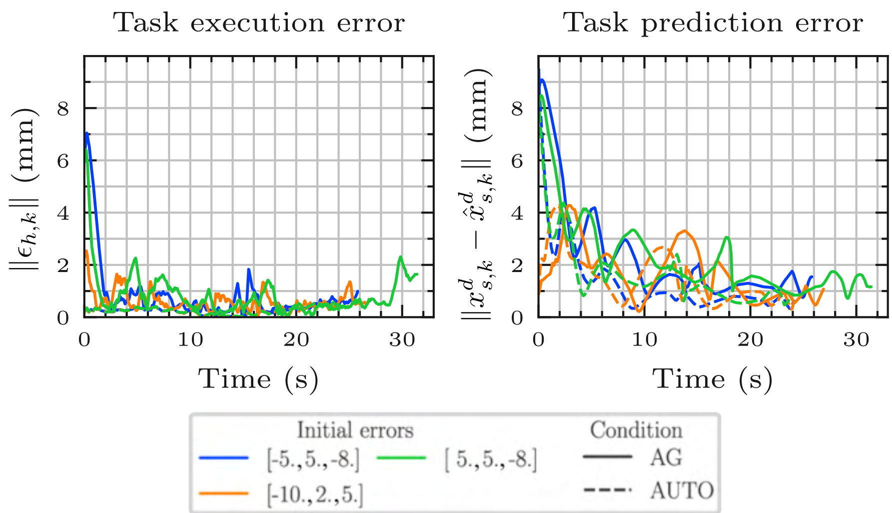


Figure 6.5: The norm of the execution (left) and prediction (right) errors under **AG** and **AUTO** are reported for different initial task parameter estimation errors $[\tilde{r}_{z,0}, \tilde{t}_{x,0}, \tilde{t}_{y,0}]$ in [deg., mm, mm]. The initial robot parameter estimation error is $\tilde{L}_0 = 10$ mm.

In order to capture overall trends in model parameter estimation errors, the mean relative parameter error w.r.t. the initial errors is computed as:

$$\tilde{\theta}_{\text{rel.},k} = \frac{1}{4}\left(\left|\frac{\tilde{r}_{z,k}}{\tilde{r}_{z,0}}\right| + \left|\frac{\tilde{t}_{x,k}}{\tilde{t}_{x,0}}\right| + \left|\frac{\tilde{t}_{y,k}}{\tilde{t}_{y,0}}\right| + \left|\frac{\tilde{L}_k}{\tilde{L}_0}\right|\right) \tag{6.2}$$

At startup, $\tilde{\theta}_{\text{rel.},k} = 1$ and if $\tilde{\theta}_{\text{rel.},k} = 0$, then all parameter estimation errors have converged to zero. The parameter errors on a_k and b_k are not included in Equation (6.2), since they tend to vary locally as the velocity imposed by the operator changes. The value of $\tilde{\theta}_{\text{rel.},k}$ is reported in Figure 6.6 (bottom), which shows that parameter estimation errors are largely reduced (by 85% or more) over the task duration. The individual parameter estimation error profiles are similar for **AUTO** and **AG**, which suggests that the learning method can effectively cope with noisy human demonstrations. Even if the human demonstrations lead to learning performances inferior to those obtained under the **AUTO** scenario, the task prediction errors also decrease under **AG**: they are reduced by more than 75% over the short execution time. The fact that a steady state is not reached for any scenario is explained by the limited duration of the task

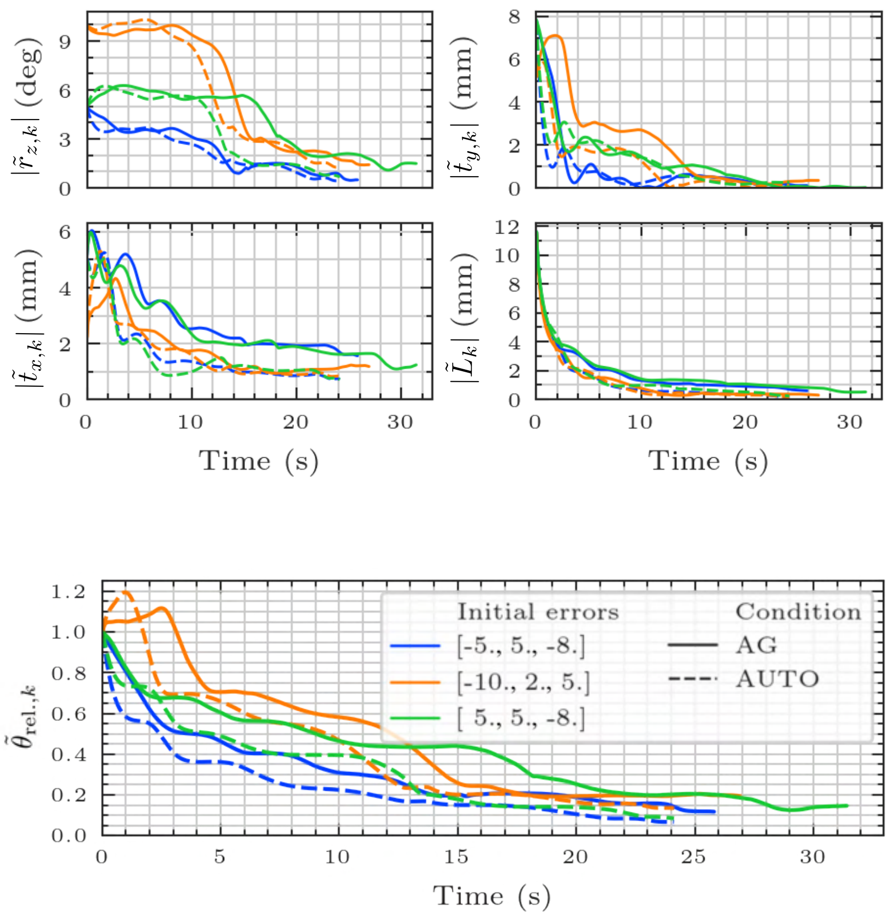


Figure 6.6: Top: parameters absolute estimation errors under **AG** and **AUTO** for different initial task parameter estimation errors $[\tilde{r}_{z,0}, \tilde{t}_{x,0}, \tilde{t}_{y,0}]$ in [deg., mm, mm]. The initial robot parameter estimation error is $\tilde{\theta}_{r,0} = \tilde{L}_0 = 10$ mm. Bottom: resulting mean relative errors $\tilde{\theta}_{\text{rel.},k}$ defined by Equation (6.2).

execution, as discussed in Section 6.3.2, but also by the geometry of the task. The downward motion from the top of the key up to the bottom (see Figure 6.4) provides very little information about the translation of the task along the direction of said motion and the associated task translation parameter $\hat{t}_{x,k}$ is barely updated over this time range (see for instance the orange curve in Figure 6.6 for $t_k \in [18; 23]$ s).

The same experiment was carried out with a variety of initial parameter estimation errors and underlying paths $\Gamma(\psi)$, including lemniscates (see Section 6.4), cursive words, and repetitive patterns. The learning method was found to be task-independent in the sense that the hyper-parameters (i.e., R_k, Q_k, P_0, and α) do not have to be changed between an experiment and another to obtain satisfactory results.

6.3.2 Influence of the fading factor

The decay time constant (20 s) associated with the chosen value for α is rather slow considering the total duration of the experiment presented above ($\approx$ 30 s). This accounts for part of the residual errors as the less reliable observations gathered at the beginning have not been totally discarded yet when the task ends. To demonstrate the influence of α on the learning, the **AUTO** condition is executed for various values of the fading factor and $\tilde{\theta}_{\text{rel.},k}$ (see Equation 6.2) is reported in Figure 6.7. The choice of α does not impact the initial updates of the parameters, but after a few seconds, the estimated parameters converge faster towards their correct values for larger values of α. This is due to the fact that the estimated covariance of the parameters is greater due to the forgetting mechanism, leading to a better learning performance when the observations are correct. The largest reported value is $\alpha = 0.01$, equivalent to a decay time-constant of 2 s such that only the last few seconds are considered for learning (see Equation 5.25).

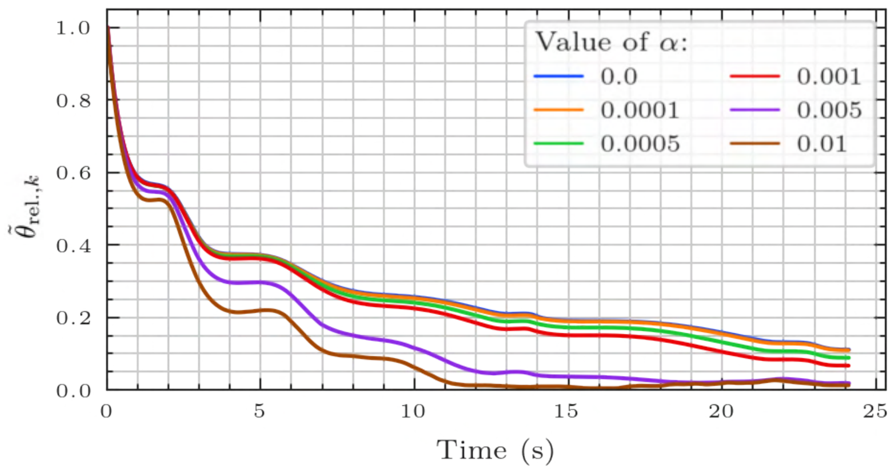


Figure 6.7: Average parameter estimation errors as a fraction of its initial value for different values of the fading factor α. The data was collected from 6 optimal executions, i.e., the **AUTO** condition.

6.3.3 Robustness to parameters shift

One of the advantages of the presented general method and its implementation with a fading memory EKF is the possibility to cope with parameters shift. As a demonstration, let us consider the same scenario: all but the path to draw is the same as the previous experiment, including the hyper-parameter values. The participant is asked to follow a path A for one revolution and then to follow a path B generated with different parameters (translation w.r.t. world frame) for the second revolution, which effectively displaces the desired path. In this scenario, our method can cope with small parameters shift as demonstrated by Figure 6.8. The task model is able to adapt to the change of path and no hyper-parameter tuning was necessary. The tracking performance could further be improved by detecting such occurrences and temporally increasing the learning rate, for example through a variable fading factor or process noise modeling.

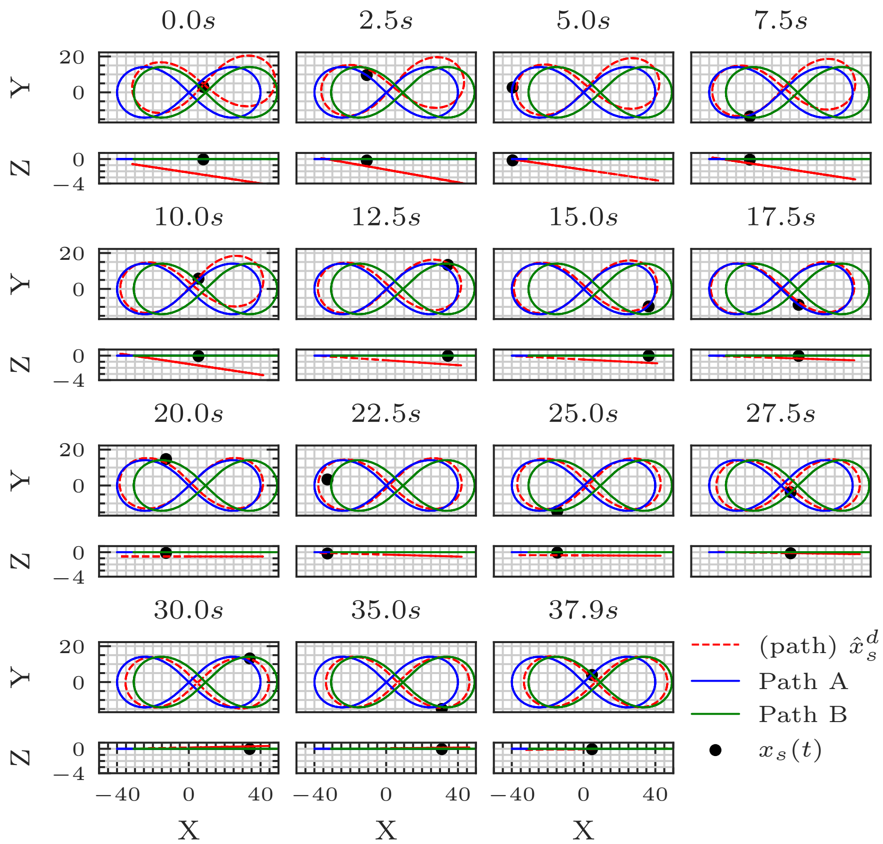


Figure 6.8: Illustrative scenario simulating a parameters shift when the first revolution is completed (around $t_k = 18$ s). Two paths A and B are visible on the drawing surface such that path B is identical to path A, but translated by 10 mm the X axis of the drawing plane. The participant follows the path A for one revolution and then path B. The task model initially converges towards path A, but is quickly adapted when a new path is followed. All distances are expressed in mm.

6.4 Comparison with non-adaptive guidance

In the previous section, we showed that user inputs can be used to simultaneously improve the task and robot models when they are initially incorrect. The online learning method was used to provide an adaptive guidance to the human operator. To further demonstrate the advantages of this approach, the task performance is evaluated in three cases: adaptive guidance, with an incorrect non-adaptive guidance, and without haptic guidance.

6.4.1 Experimental protocol

The context is the same as the one detailed in the previous section: the model of the path to follow is known, but its registration is partially unknown. Initially, only rough estimates of the parameters θ_g and θ_r are available. Under condition **AG**, a human operator is teleoperating the robot in order to perform the task and haptic guidance is provided. Two additional experimental conditions are considered:

IG A human user teleoperates the robot in order to perform the task and haptic guidance is provided. The models of the task and of the robot are NOT corrected (incorrect guidance);

NG A human user teleoperates the robot in order to perform the task and no haptic guidance is provided (no guidance). However, the learning method is used to learn the parameters.

Under condition **IG**, only the tracking of the parameters a and b is performed to accommodate the execution velocity imposed by the operator. The other four parameters, encoding the task and tool partial registration, are not updated such that $\theta_k = \theta_{g,k} = [a_k,\, b_k]^T \in \mathbb{R}^2$. This condition can be considered as a normal virtual fixture whose registration is inaccurate and behaves as existing (non-adaptive) haptic guidance methods, such as Guidance VF (Bowyer et al. 2014). The only particularity is that, due to the proposed method, the guidance reference is not simply computed as the closest point on the guidance path, but from the estimated parameters a_k and b_k. This tracking introduces a dynamics in the VF that is experienced by the operator as inertia in the direction tangent to the guidance path, which tends to smooth the execution by filtering accelerations. Under the two other conditions, the proposed online learning method is used to update the initially incorrect parameters from the observed user actions. What differentiates the conditions **AG** and **NG** is how the models are used. The former provides a haptic guidance whose underlying model is refined as the parameters are updated whereas the later passively observes without providing any haptic guidance.

Six participants were recruited for this experiment. They were asked to teleoperate the robot to draw a lemniscate by following a path printed on the drawing surface (see figures 6.1c and 6.9). In order to give the learning algorithm enough time to reach a steady state, the participants were asked to perform two complete revolutions for each experimental condition. Before performing these tasks, they were allowed to practice by drawing lines and circles without and with haptic guidance (correct models) for at least 5 minutes. Then, they performed the actual task under each condition in a balanced pseudo-random order (single blind experimental protocol) for a total of 3 task executions per participant. The visual feedback and evolution of task models for one of the participants is included as supplementary material for illustration purposes: see Video 6.1 for **AG**, Video 6.2 for **IG**, and Video 6.3 for **NG**. The learning performance for six correct automatic executions (i.e., **AUTO** condition) was also added to provide a baseline performance. The experimental setup, learning and guidance methods, and all hyper-parameters are the same as those presented in the Section 6.4.1.

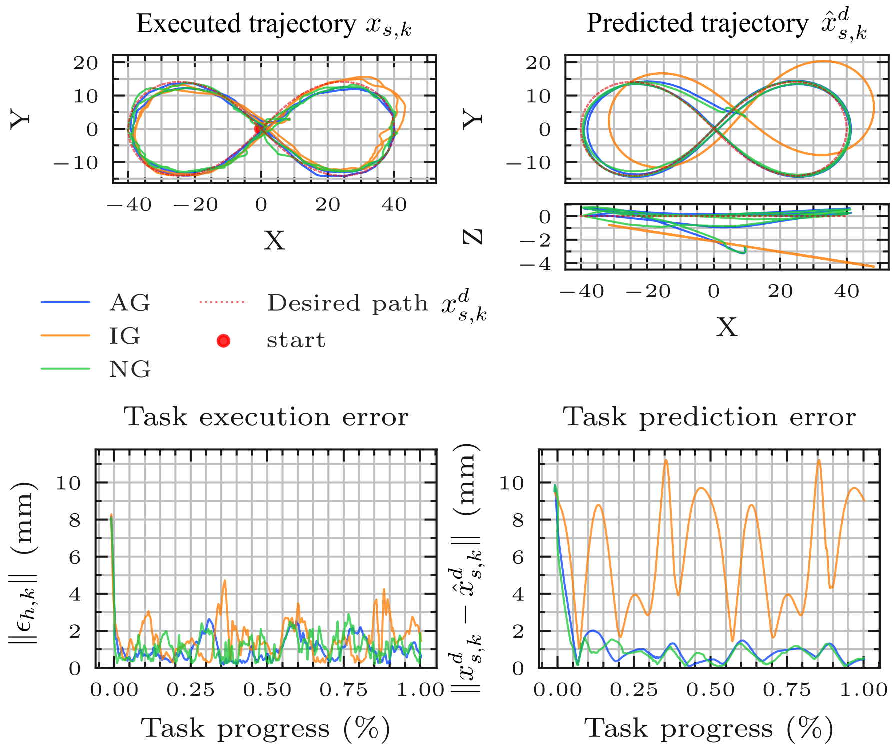


Figure 6.9: Top: executed, predicted, and desired trajectories are displayed as paths in the drawing plane XY and projected orthogonally (XZ plane) for the three task executions performed by one of the participants. Bottom: associated errors. To facilitate the reading and to cope with disparities in execution velocities, all values are expressed w.r.t. the task progress defined as the normalized curvilinear abscissa such that the task execution starts at 0% and ends at 100%. Thus, the first revolution around the lemniscate is over when the task progress reaches 50%. All of the values are expressed in mm.

6.4.2 Results

When learning from optimal demonstrations (**AUTO**), the average relative parameters error is reduced by 95% by the end of the first revolution that corresponds to 20 seconds of task execution (see Figure 6.10). In this scenario, the average relative parameters error converges to very small values (2%) such that the task and robot model prediction errors were under 0.5 mm by the end of the task execution (see Figure 6.11). When learning from human demonstrations (**AG** and **NG**), the average relative parameter estimation errors are also very significantly reduced by the end of the first revolution (by 90%). The residual state parameter estimation error is larger than for the **AUTO** scenario, in the vicinity of 5% for both **AG** and **NG** (see Figure 6.10)

The steady state of the learning process is reached, on average, for a task progress of 70% or 35-40 s of task execution. To characterize the residual errors, different metrics are computed for the last 20% of the task and averaged across subjects. The residual task prediction error is around 0.5 mm for both **AG** and **NG**, respectively 0.52 ± 0.32 mm (max 0.86) and 0.56 ± 0.21 mm (max 0.83), where the number following the "±" symbol is the standard deviation across participants. The residual robot model prediction error is 0.25 ± 0.19 mm for **AG** and 0.28 ± 0.15 mm for **NG**. For comparison, under the condition **AUTO**, the residual task and robot model prediction errors are, respectively, 0.19 ± 0.003 mm and 0.22 ± 0.01 mm. For this same last 20% of the task execution, the mean relative parameter error w.r.t. initial ones are $5.88 \pm 2.37\%$ for **AG**, $6.32 \pm 1.66\%$ for **NG**, and $2.1 \pm 0.02\%$ for **AUTO** (see Figure 6.10).

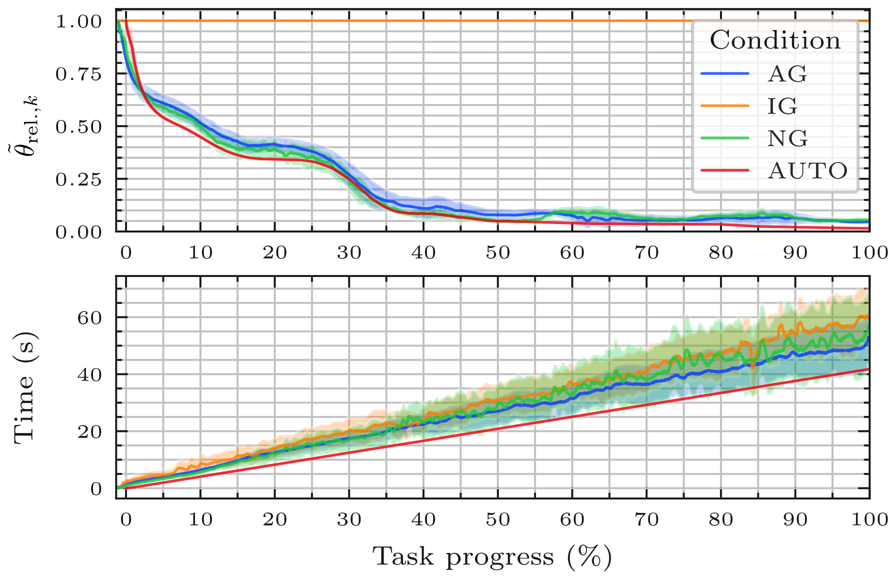


Figure 6.10: Top: the mean relative parameter estimation error $\tilde{\theta}_{\text{rel.},k}$ (see Equation 6.2) is averaged over the participants (N=6) for each condition. The same is done with 6 automatic executions of the scenario. Bottom: the correspondence between task progress and execution time is displayed. Standard deviation across participants is displayed as a shaded region.

The mean prediction and execution errors are computed for the full task duration across subjects for each condition (see Figure 6.12). The mean task prediction errors during **AG** (1.07 ± 0.24) and **NG** (1.03 ± 0.14) are significantly lower ($p < 0.001$) than under **IG** (6.03 ± 0.17). The difference in mean task prediction errors between **AG** and **NG** was found to be statistically insignificant ($p \approx 1$). Similar results are found for the mean robot model prediction error that is of 11.0 ± 0.0 for **IG** against 0.5 ± 0.06 and 0.54 ± 0.13 for **AG** and **NG** respectively. The results are comparable between **AG** and **NG** ($p \approx 1$). The analysis of task execution errors shows that a

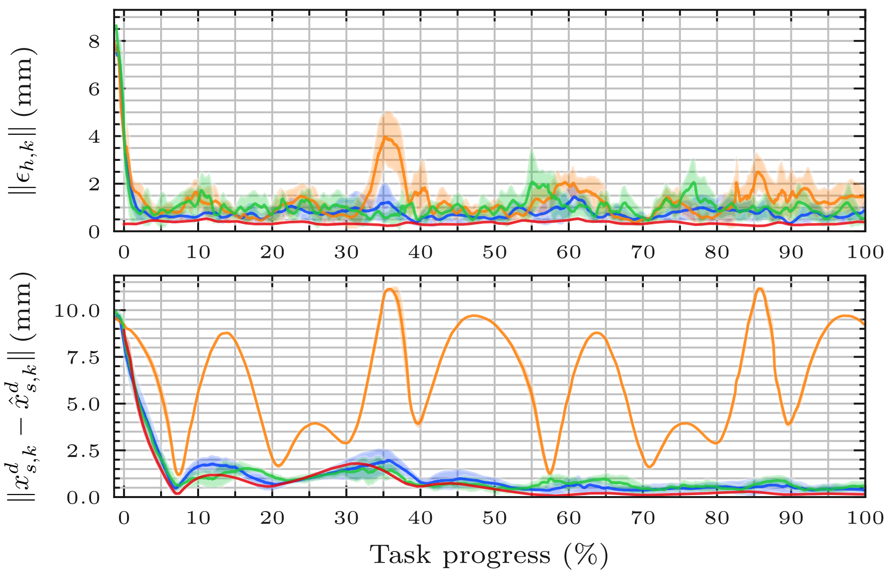


Figure 6.11: Mean execution (top) and prediction (bottom) errors for each experimental conditions. The mean values are displayed with solid lines and the standard deviation ($N = 6$) with a shaded region. See Figure 6.10 for the legend, Figure 6.9 for details for one participant, and Figure 6.12 for overall scores.

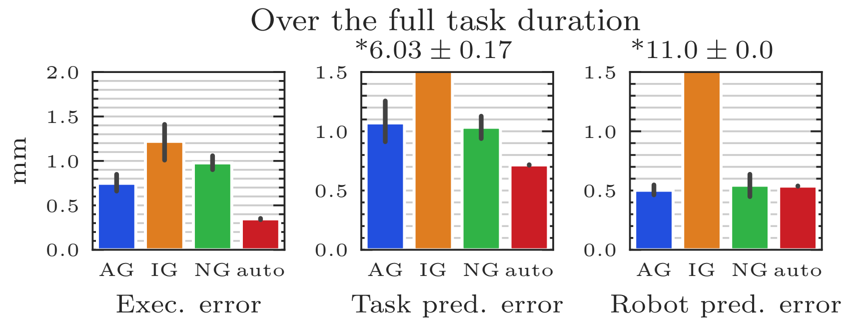


Figure 6.12: Mean norms of the execution, task prediction, and robot model prediction errors across subjects in mm. The standard deviations are displayed as black segments. Prediction errors for the **IG** condition are too large to fit in the figure and their means and standard deviations are displayed above the bar.

correct haptic guidance leads to significantly ($p < 0.005$) lower errors (**AG**, 0.75 ± 0.13) than both when the guidance is incorrect (**IG**, 1.22 ± 0.29) or when there is no haptic guidance (**NG**, 0.98 ± 0.11). For reference, the average execution error under **AUTO** is 0.35 ± 0.01 mm. Although no guidance seems to be better than an incorrect guidance in these experiments, it was not found to have a significant effect ($p > 0.1$).

6.4.3 Analysis

In this experiment, the automatic execution of the correct task (**AUTO**) results in a mean execution error of 0.35 mm. The task prediction error is then considered to be negligible below this value. Under this threshold, the residual errors are due to a combination of intrinsic robot positioning inaccuracies and, more importantly, ground truth registration inaccuracies (e.g., deformation of the pen tip or body and slight orientation estimation errors during the offline calibration). When learning from optimal demonstrations (**AUTO**), the estimated task and robot models are then found to be as accurate as the offline models identification once the steady state is reached. Due to execution errors and biases introduced by the participants, this level of accuracy is never reached under **AG** and **NG**, but the prediction errors are reduced to satisfactory levels under 0.5 mm (the width of the line drawn by the pen is $0.5 - 0.7$ mm wide and the task is 80 mm wide). The proposed method is then capable of retrieving the correct task and robot model parameters online by exploiting only the teleoperation data provided by the presence of an operator *in-the-loop*.

Updating the haptic guidance through online model corrections allows to improve the overall performance when the models are initially incorrect. Compared with a non-adaptive virtual fixture (**IG**), using adaptive guidance (**AG**) led to a task execution error reduction of 40%. The results suggest that providing no guidance or providing an inaccurate, not updated, guidance leads to similar performances.

It was initially supposed by the authors that improving the haptic guidance over time (**AG**) would make task execution better over time, and in turn would improve parameters estimation. It was shown that **AG** decreased execution errors, but the presence of haptic guidance does not have a significant impact on the overall learning performance as suggested by figures 6.10 and 6.11. This was confirmed by the mean task and robot model prediction errors not being significantly different under **AG** and **NG**, both for the complete tasks (see Figure 6.12) and once the steady state is reached. Interestingly, this means that the models could also be learned online even though they are not used to physically guide the user, for instance to provide other types of information such as visual guidance by displaying the corrected plan with augmented reality.

6.4.4 Necessity for joint robot and task models corrections

In order to demonstrate why we have to correct the follower robot kinematic model in addition to the task model, we carried out the same experiments as presented in Section 6.4 without updating the robot model parameter. The results under this additional scenario **AG-TO** (AG Task Only) are reported in Figure 6.13, where the predicted desired trajectory, displayed as a path in the XY and XZ planes, clearly does not converge towards the actual desired trajectory. Some of the underlying parameters, namely the orientation $\hat{r}_{z,k}$ and lateral translation $\hat{t}_{y,k}$, partially converge towards the correct values, but significant errors remain for the other parameters. Notably, $\hat{t}_{x,k}$ converges towards an inaccurate value as clearly visible in Figure 6.13. The guidance errors are not limited to the direction normal to the drawing plane, as the prediction errors are also very significant in the XY plane (see Figure 6.13). The prediction error is then on par with the non-adaptive **IG** condition, with a mean task prediction error of 8 mm overall and of 7.3 mm over the last 20% of the task.

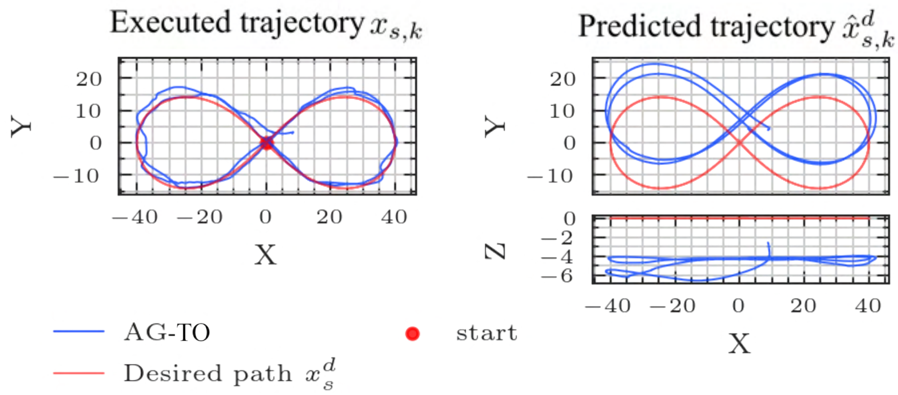


Figure 6.13: Executed, desired, and predicted trajectories for the **AG-TO** scenario. All units for X, Y, and Z are in mm.

6.5 Conclusions

In Chapter 5, we proposed a flexible approach to use information provided by an operator *in-the-loop* to augment available information for online task and robot kinematic model parameters learning. The approach is applicable when models of the robot and of the task are known, but that the values of some underlying parameters are not. This is typically the case when a task has been defined but parameters have to be adapted to the actual environment during its execution. Such parameters include, but are not limited to, task and robotic tool registration parameters.

In this chapter, an implementation on a robotic platform with human participants was provided to demonstrate the applicability of such an approach to simultaneously correct a pre-planned trajectory and a nominal robot kinematic model from teleoperation data only. The results further demonstrate that the learning can cope with small variations of the correct parameters, which is of great interest for applications such as robot assisted surgery where, due to environment deformations, the correct task model might change during the execution of the task. As the learning is done online, while the operator is performing the task, the updated parameters can then be used to provide an adaptive assistance that improves over time. In a scenario with initially incorrect task registration and robot tool calibration, the proposed adaptive haptic guidance significantly improved task execution accuracy compared to a non-adaptive haptic guidance.

Chapter 7

Qualitative performance of the adaptive haptic guidance

Contents

7.1 User-study design . . . 112
7.1.1 Experimental protocol . . . 112
7.1.2 Experimental setup . . . 114
7.1.3 Clutching for mapping re-calibration . . . 114
7.1.4 Collected experimental data . . . 115
7.2 Results . . . 117
7.2.1 Objective results across participants . . . 117
7.2.2 Subjective results across participants . . . 118
7.3 Analysis . . . 119
7.3.1 Effects of inaccuracies . . . 119
7.3.2 Performance and acceptance of adaptive haptique guidance . . . 120
7.3.3 Perception of guidance compliance . . . 121
7.4 Effect of the mapping recalibration . . . 121
7.5 Conclusions . . . 123

In the previous two chapters, we proposed and evaluated a method for simultaneous task and robot model parameters learning. Results showed that the models could be accurately learned from a human in-the-loop and that the adaptative guidance improved the accuracy of the drawing task. However, some practical considerations were not taken into account in this first experimental validation.

Firstly, the drawing task was roughly the size of the master workspace such that it could be performed as a single continuous motion. However, in most real-world telerobotics applications, including robot-assisted MIS, the range of master motions is insufficient to cover the whole operational space of the follower robot. Secondly, the adaptive guidance scheme was not compared with a haptic guidance generated from correct models. Finally, only objective metrics were considered (i.e., task execution accuracy), but subjective performance is critical to the acceptance of haptic guidance by operators.

Therefore, we report more thorough qualitative results concerning the effect of registration inaccuracies on haptic guidance while assessing how online learning can improve the performance in such cases. The task spans a large space such that clutching, a mapping recalibration approach commonly used in telerobotics, is necessary to adjust the mapping between master and follower robot during the realization of the task. We also introduce an additional experimental condition for which the models used to generate the haptic guidance are correct. Both objective and subjective results are reported, all collected during a study with twelve participants. Then, in a second section, a preliminary analysis of the impact of mapping clutching on teleoperation with haptic guidance is provided.

7.1 User-study design

An online method was proposed in Chapter 5 to correct initially inaccurate task and robot kinematic model parameters from operator inputs. In this chapter, we evaluate the effects of these inaccuracies on the performance of a path following task when haptic guidance is provided. Although the robotic setup is the same (see Figure 6.1), the task spans a larger volume of the follower workspace such that a mechanism to modify the position teleoperation mapping $\mathcal{M}(\cdot)$ is necessary. The task is also longer (i.e., total underlying path arc-length) and therefore takes more time to perform. In addition, while the experimental validation in Chapter 5 focused on the convergence of the parameters and the evolution of errors resulting from inaccurate modeling, the quality of the guidance and executed motion was not evaluated by the participants, whereas it will be in the following.

7.1.1 Experimental protocol

Twelve participants, all right handed, were recruited as part of a study approved by an ethical committee [1]. After being given adequate information about the material and asked to fill the consent forms, each participant practiced 5 minutes with and

[1]University of Strasbourg, France. Accreditation n° **Unistra/CER/2022-09**.

without haptic guidance by drawing a circle. They then performed a wide motion (e.g., line) to practice clutching and finally performed free form drawings. Once the training phase was finished, each participant performed a task under 4 experimental conditions. They started the task when engaging the position control by pressing a master push button, reached the startup position, and then followed the path to its end under visual feedback (see Figure 7.1). The same button, when released, triggered the clutching of the mapping as detailed afterwards. The visual feedback recorded during the realization of the task by one of the participants (under **AG** condition) is included as complementary material (see Video 7.1).

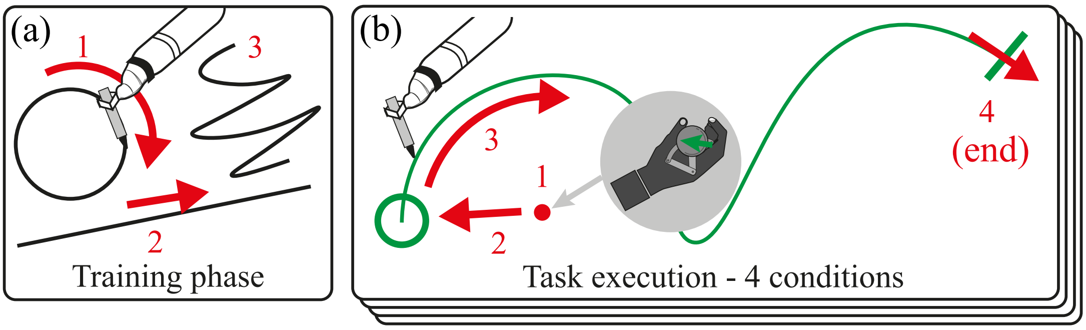


Figure 7.1: Overview of experimental protocol. (a) Participants are asked to practice 5 minutes with and without haptic guidance. (b) Then, they perform the same task under 4 experimental conditions and fill a questionnaire at the end of each condition.

In order to compare the effect of model inaccuracies and online corrections, both adaptive and non-adaptive guidance were implemented, as well as a scenario where no guidance at all was provided. With the exception of an additional non-adaptive, but correct haptic guidance, the experimental conditions are the same as those presented in Chapter 6:

- **NG** – no guidance forces are applied on the master interface;
- **IG** – A non-adaptive guidance is provided, but the model parameters are incorrect (see Section 6.4.1);
- **CG** – A non-adaptive guidance is provided and the model parameters are correct. This condition is identical to **IG**, but with correct model parameters;
- **AG** – Adaptive guidance as proposed in chapters 5 and 6.

Additionally, the task is automatically executed 12 times to have a baseline as was done in Chapter 6. This last condition is referred to as **auto**.

Throughout the experiments, we tested further our main hypotheses than can be stated as:

- **H1** – when haptic guidance is provided, if the models of the task and/or of the robot are inaccurate, the execution errors will be similar or larger than if no guidance was provided;

- **H2.1** – although imperfect, human actions can be used to learn the correct task and robot models;
- **H2.2** – by correcting the models online, conflicts are significantly reduced, which leads to better performances overall in comparison to a non-adaptive guidance.

An across-subjects experimental protocol was devised, such that all participants executed the task under all conditions that were presented to them in a pseudo-random balanced order. After completing each task under a particular condition, each participant filled a NASA-TLX (Hart et al. 1988) questionnaire to evaluate subjective preferences between conditions. The questions assess arduousness through items such as perceived frustration and cognitive load (see questionnaire in Appendix D.2). In addition to the NASA-TLX questionnaire, two questions were included to evaluate if the participants found the haptic guidance to be useful (see Table 7.1). As the second question (Q2) requires the haptic guidance to be activated, participants were told to ignore this question when performing the task under **NG** condition.

	Question	Likert scale (5 points)
Q1	I felt in control while performing the task.	1: Strongly disagree 2: Disagree
Q2	The forces generated by the master interface helped me to perform the task.	3: Neither agree nor disagree 4: Agree 5: Strongly agree

Table 7.1: Additional survey questions for each condition. Translated from French, see original in Appendix D.2.

7.1.2 Experimental setup

The task to perform (see Figure 7.2) and the experimental protocol differ from those in Chapter 6, but the underlying implementation of the adaptive haptic guidance is the same. The haptic guidance is implemented with a stiffness of 300 $N.m^{-1}$ and a damping ratio of 0.9. The fading memory EKF is implemented with a constant fading factor $\alpha = 10^{-3}$ (20 s decay time constant) and an isotropic Gaussian model of standard deviation $\sigma_h = 2$ mm for the modeled human inaccuracies (see $\Sigma_{h,k}$ in Chapter 5). The noise modeled on the execution velocity imposed by the operator is $\sigma_{\dot{\psi}} = 0.1 \text{mm.s}^{-1}$. The initial parameter covariance matrix is initialized with small values encoding an initial standard deviation of 10 mm for $\hat{a}$, 10 mm.s^{-1} for $\hat{b}$, 1 deg. for the in-plane task orientation, and 1 mm for the three remaining task and robot parameters.

7.1.3 Clutching for mapping re-calibration

When the follower robot operational workspace is significantly larger than the master robot operational workspace, the teleoperation implementation requires special care. In order for the operator to be able to execute a task spanning a significant part of the follower robot operational space, one of four approaches can be considered:

- a scaling can be applied such that a small master displacement results in a large follower displacement (Kobayashi et al. 1992), but this also scales the errors introduced by the operator;
- rate control can be used to map the position of the master to the follower robot commanded velocities (Horan et al. 2008);
- mapping clutching (Niemeyer et al. 2016) can be implemented such that the operator can manually adapt the mapping between master and follower position by repositioning the master independently from the follower when clutched (i.e., a button is pressed);
- mapping drifting consists in adapting the mapping continuously such that the master robot stays in the center of its workspace (Conti et al. 2005; Dominjon et al. 2007).

For surgical robotics, a scaling factor greater than 1 is not a viable option as it would increase execution errors when the opposite is usually desirable (e.g., tremor suppression Prasad et al. 2004). Rate control is also excluded since the position mapping is crucial to dexterous gestures and would result in a higher cognitive load during trajectory-following tasks. Therefore, the only options considered thereafter are mapping clutching and drifting. Workspace clutching is a technique commonly used in teleoperated surgical systems (e.g., Da Vinci) and more generally for position-position (or, equivalently, velocity-velocity) teleoperation. We considered that mapping drifting would introduce more complex effects that could interact with the adaptive guidance. Therefore, mapping clutching was implemented. In practice, it was implemented by modifying the mapping $\mathcal{M}(\cdot)$ such that $\mathcal{M}(x_m)$ stays constant during the so-called clutching phases, therefore generating no follower robot displacement.

7.1.4 Collected experimental data

The collected experimental data for one participant under the 4 conditions is displayed in Figure 7.2. Under **IG** and **AG**, the task model registration is incorrect when the task execution starts, but it is refined under **AG** such that the task model is correct after 20 to 30s (see snapshots in Appendix D.1), which is 15% to 25% of the average total task duration. The executed ($x_{s,k}$) and predicted ($\hat{x}^d_{s,k}$) trajectories are recorded and compared with the ground truth task model (i.e., desired trajectory $x^d_{s,k}$) to compute the execution errors $||\epsilon_{h,k}||$ and task prediction errors, respectively. Likewise, the robot estimated trajectory $\hat{x}_{s,k}$ from the robot model is recorded and compared with executed trajectory $x_{s,k}$ to compute the robot prediction errors. Additionally, the total execution time, or task duration, is recorded and the smoothness of the executed follower trajectory is evaluated using the so-called spectral arc-length metric (Balasubramanian et al. 2015).

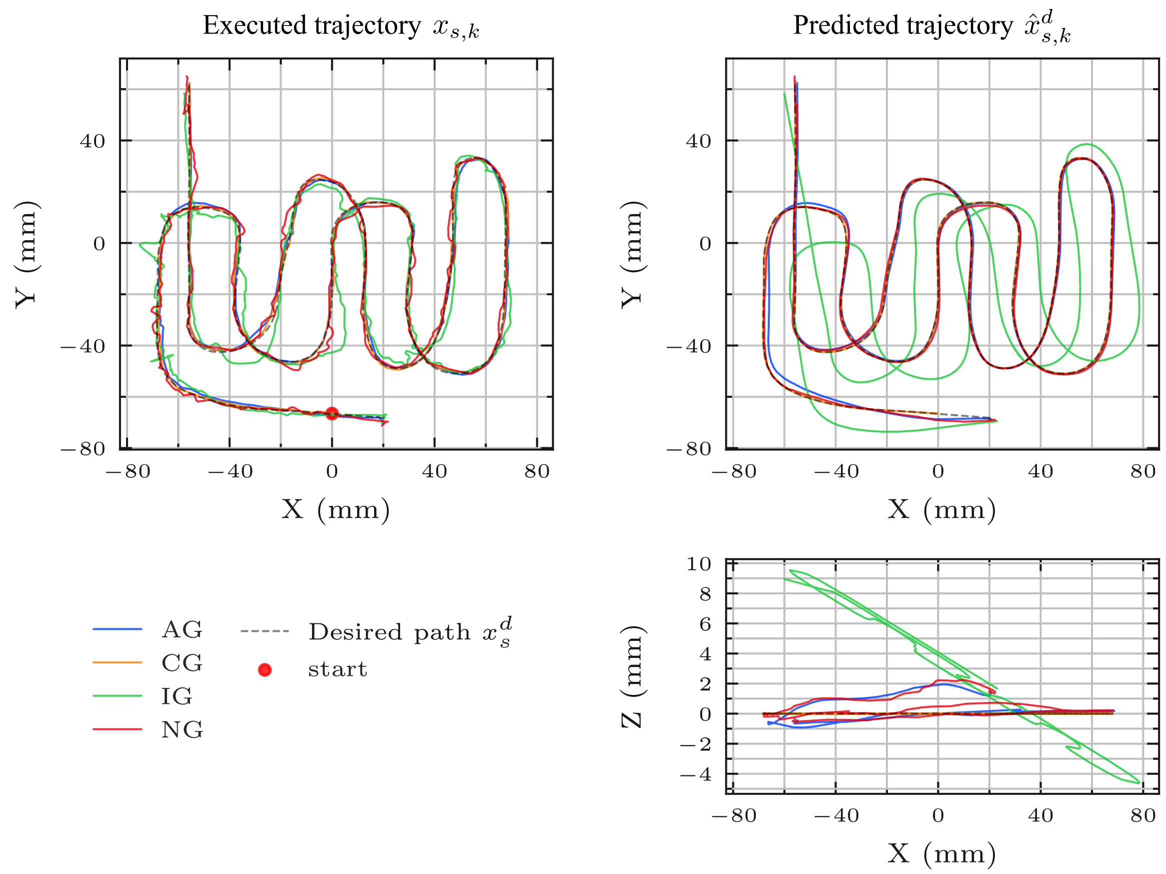


Figure 7.2: Path underlying the trajectories $x_{s,k}$ and $\hat{x}_{s,k}$, respectively the executed and predicted trajectories, expressed in $\mathcal{F}_t$ for one of the participants under the different experimental conditions. See figure Appendix D.1 for snapshots at different time points (**AG** condition).

7.2 Results

7.2.1 Objective results across participants

The results obtained during this experiment (see Figure 7.3) are in agreement with those from Chapter 6:

- task execution errors under **AG** are significantly lower than under **NG** and **IG**;
- task prediction errors are significantly reduced by the learning process (**AG** and **NG** against **IG**);
- robot prediction errors are also significantly reduced by the learning process (**AG** and **NG** against **IG**);
- there is no statistically significant difference between the prediction errors, both task and robot, under **AG** and **NG**.

Additionally, the analysis of the **CG** condition led to the following statistically significant differences. As can be expected, task and robot prediction errors are significantly smaller under **CG** than under **AG**, **NG**, or **IG**. Furthermore, task execution errors are significantly smaller under **CG** than under **NG** or **IG**. However, there is no statistically significant difference between the execution errors during **AG** and **CG**. Finally, the total task duration (about 2 minutes on average) was not found to be significantly impacted by the experimental condition.

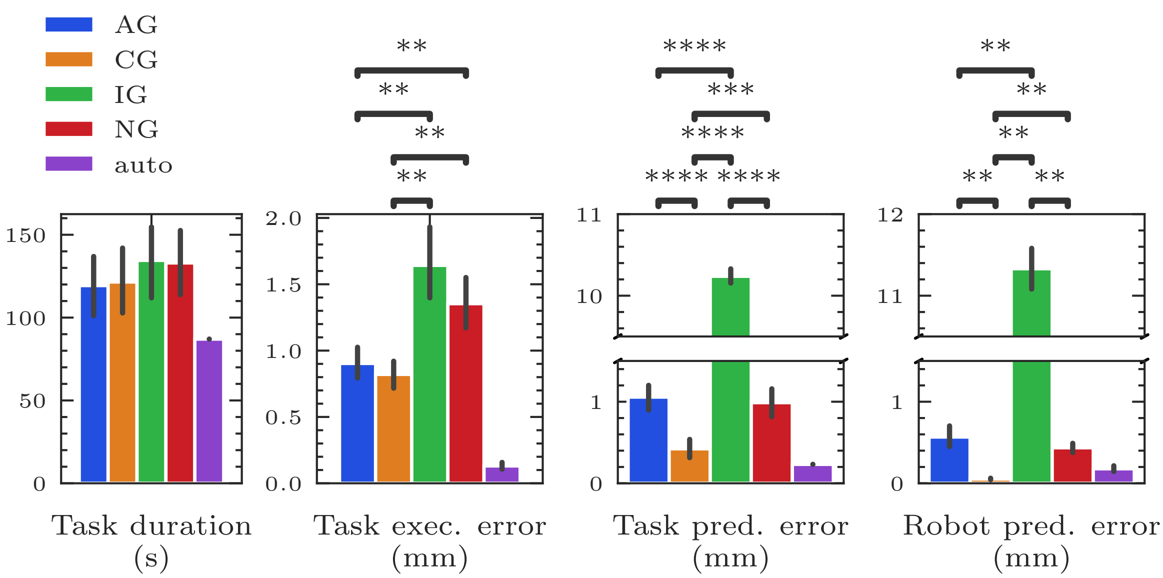


Figure 7.3: Task duration, execution errors (path deviation), and task and robot model prediction errors ($N = 12$). Significance is reported ; the levels are ns (p> 0.05, not represented), $*$ (p< 0.05), $**$ (p< 0.01), $***$ (p< 1.10^{-3}), and $****$ (p< 1.10^{-4}).

The smoothness of the executed robot motion was also investigated. The spectral arc length of the follower Cartesian velocity profile was computed for each participant under each condition using the SPARC library (Balasubramanian et al. 2015). The method was implemented with a cut-off frequency of 10 Hz and a threshold of 5.10^{-2} for signal amplitude. As would be expected, the automatically generated motion (**auto**) is much smoother than the ones performed by the participants. Interestingly, the results show that the motion is significantly less smooth under **NG** than under any other condition (see Figure 7.4). The only other significant result is that the motion is significantly smoother under **AG** than under **IG**. It can also be noted that **AG** and **CG** have comparable scores.

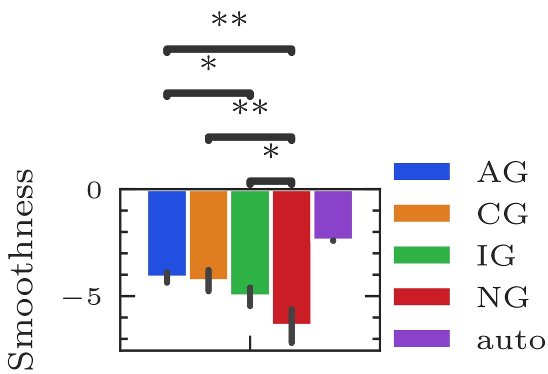


Figure 7.4: Motion smoothness for the different conditions. The metric is the spectral arc length and larger values indicate smoother movement (Balasubramanian et al. 2015).

7.2.2 Subjective results across participants

The 6 individual NASA TLX scores and the resulting global score (unweighed mean of the 6 individual NASA TLX scores) are averaged to compute subjective metrics across the 12 participants (see Figure 7.5). Overall, the participants significantly preferred the conditions **AG** and **CG** over the condition **IG**. They also significantly preferred **AG** over **NG**. No other difference was found to be statistically significant. However, we also found several significant effects of the experimental condition on the individual scores (see Figure 7.5):

- the participants found that the **IG** condition resulted in significantly more Effort and Frustration than the conditions **AG** and **CG**. Additionally, the **NG** condition resulted in significantly more Effort and Frustration than **AG**;
- the same differences can be noticed in the performance as evaluated subjectively by the participants. Participants found their Performance to be significantly better during conditions **AG** and **CG** compared to both **NG** and **IG**;
- participants found the condition **AG** to be significantly less mentally demanding than the condition **NG**;

- no other difference was found to be statistically significant.

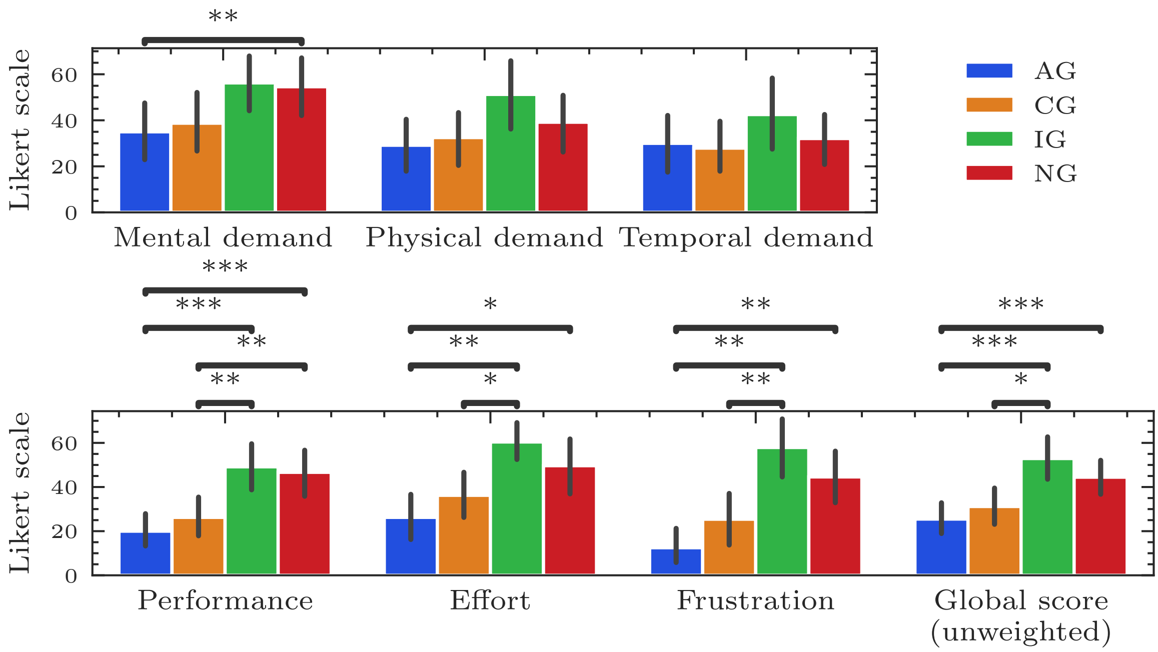


Figure 7.5: NASA TLX scores (lower is better). Significance is reported ; the levels are ns (p> 0.05, not represented), $*$ (p< 0.05), $**$ (p< 0.01), $***$ (p< 1.10^{-3}), and $****$ (p< 1.10^{-4}).

The two additional questions that the participants had to answer after each experimental conditions were also analyzed for statistically significant effects (see Figure 7.6). Participants reported they felt more in control (Q1) under **AG** and **NG** conditions w.r.t. the **IG** condition. They also more strongly agreed with the statement Q2 under **AG** and **CG** conditions w.r.t. the **IG** condition, indicating that they found the haptic guidance significantly more helpful under **AG** and **CG**.

7.3 Analysis

7.3.1 Effects of inaccuracies

The performance under **CG** condition was found to be significantly better than under condition **IG**. Not only are the execution errors lower and the movements smoother under **CG**, but it was also experienced significantly more positively by participants. They found the **IG** condition to be more demanding, more frustrating, and resulting in a perceived decrease of performance. These quantitative results heavily support the first part of hypothesis **H1**, which states that an inaccurate guidance will increase execution errors and decrease performance.

We had initially hypothesized in **H1** that condition **IG** would lead to performances similar or worse than **NG**. In Chapter 6, there was no significant difference in objective performance between **IG** and **NG**, suggesting that performances (execution errors only

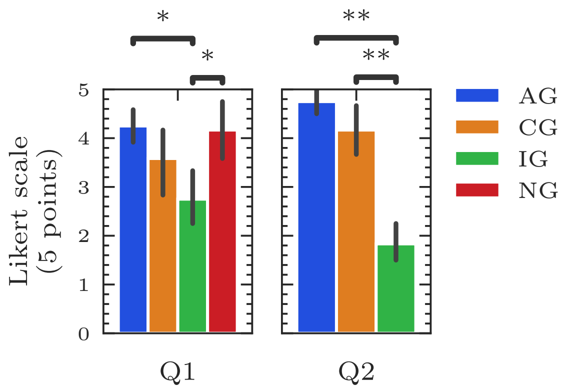


Figure 7.6: Scores obtained for the two additional survey questions (see question statements in Table 7.1).

in that case) were similar. In this study with more participants, the level of smoothness was found to be significantly different between **IG** and **NG**: the movements under **IG** were smoother ($p < 0.05$). However, this is the only significant effect we found in objective metrics. As for the subjective metrics, only the answers to Q1 were found to be significantly different ($p < 0.05$), suggesting that participants felt more in control under **NG**. Although not statistically significant, one can also see a trend in the NASA TLX scores (see Figure 7.5) in favor of **NG** over **IG**.

To conclude, inaccuracies in the models used to generate guidance (i.e., **IG**) significantly degrade performance. Although such inaccurate guidance still improved movement smoothness compared to a non-assisted teleoperation (**NG**), no guidance was overall preferred by the participants. They notably felt more in control without the guidance than with an inaccurate one. However, a correct guidance (**CG**) led to significantly better performances (accuracy, smoothness, and subjective metrics) than both the inaccurate guidance (**IG**) and the absence of guidance (**NG**).

7.3.2 Performance and acceptance of adaptive haptique guidance

Hypothesis **H2.1** was already supported by the results presented in Chapter 6: correct parameters can be learned from the actions of the operator, even when the initially incorrect parameters are used to guide the operator or when no guidance is generated. The data collected during the study with participants confirmed these results. Similarly, the experimental results support **H2.2** that states that **AG** would lead to significantly better performance than **IG**. Compared with the non-adaptive guidance (**IG**), the adaptive guidance scheme (**AG**) significantly improves the performance and was largely preferred by the participants. We did not find a single statistically significant performance difference between **AG** and **CG**, suggesting that the performances under each conditions are similar.

The overall trend is that **AG** and **CG** performed similarly, and better than both **NG** and **IG**. Under condition **AG**, movements were found to be more precise and smoother than under both **IG** and **NG**. Furthermore, although participants found the guidance unhelpful under **IG** (see Q2 in Figure 7.6), they all found the guidance helpful under

AG. Furthermore, **AG** was ranked as well as **CG** and significantly better than **IG** (and **NG**) in the NASA TLX questionnaire.

7.3.3 Perception of guidance compliance

Although not statistically significant, there is a trend in the subjective results that the condition **AG** is perceived more positively than the condition **CG**. This was not expected, but as it was noticed early on in the study, the participants were asked some questions at the end of their participation beyond the ones in the questionnaires, including whether they preferred **CG** or **AG** and why. Although most had no preference, some of the collected answers showed slight preference for **AG** because it was "more compliant" and some mentioned a phenomenon for **CG** that they described as "damping" along the path.

This "damping" phenomenon is normal and explained by the tracking of ψ_k, whose dynamics tend to smooth the gesture, which is experienced as forces tangent to the planned path by the operator. However, it is present under all condition except **NG**, such that it does not explain the participants feedback. After investigation, it was found that the guidance under the **AG** condition could indeed be more compliant concerning the motion along the path due to a larger covariance, in part due to additional human path deviation and the interactions of the temporal registration with the other parameters. This could explain the fact that **CG** was given a slightly higher level of perceived Frustration and Effort on the NASA TLX questionnaire (see Figure 7.5). However, note that the difference remains statistically insignificant.

Another explanation is that the guidance path updates, and therefore the experienced compliance of the guidance are perceived positively: at startup, under **AG**, the participants can feel that the guidance is incorrect, but then also sense that it is improving over time. Such biases could be further investigated by implementing haptic guidance in such a way that the uncertainty of the parameters does not impact the tracking of ψ and that there is no sensation of damping (like a classical virtual fixture). It should, however, be noted that the damping along the path introduced by the filter improves numerical stability and smooths the guidance, especially when the underlying path is incorrect and in the presence of strong curvature.

7.4 Effect of the mapping recalibration

During the mapping recalibration phases (clutching), the haptic guidance forces are still generated on the master side since the guidance reference $x^g_{m,k}$ is translated along with the master position $x_{m,k}$. Guidance forces could be turned off, for example using variable impedance control to set the stiffness to zero. Although variable impedance could lead to a loss of passivity, such issues can be avoided using so-called passivity filters (Bednarczyk et al. 2020) or passivity tanks (Ferraguti et al. 2013). However, it should be noted that turning the guidance on and off in such a way would still generate sudden force variation over short time periods that could be experienced as a disturbance by the operator. That is why the forces were kept on during the recalibration phases in the work presented in the previous sections.

To the best of our knowledge, very few studies on the effects of clutching during teleoperation with haptic feedback were reported. A comparison of rate control teleoperation against velocity-velocity (with clutching) teleoperation was reported by Abi-Farraj et al. (2018) for a grasping task. The authors concluded that rate control was more demanding (i.e., higher cognitive load) and that the participants preferred the velocity-velocity teleoperation. However, the clutching process was experienced negatively by the participants. The general consensus is that clutching has a negative impact on performance, but the effects are rarely analyzed, if ever. In the following, we present preliminary results on whether or not mapping clutching, during teleoperation with haptic guidance, has a negative impact on the task execution accuracy.

Firstly, it can be observed that the experimental conditions do not seem to impact the way participants use the mapping clutching (see Figure 7.7). Under all conditions, the number of clutch events, as well as their duration (about 2 s) are similar. Although no significant effect was found (ANOVA analysis resulted in p-value $p = 0.55$), the participants tend to spend more time re-calibrating the mapping under **IG** (about 35% more than under **AG** and **CG**). Then, whether the haptic guidance is correct, incorrect, or even absent does no seem to significantly affect clutching patterns. This does not, however, inform on whether mapping clutching impacts performance or not.

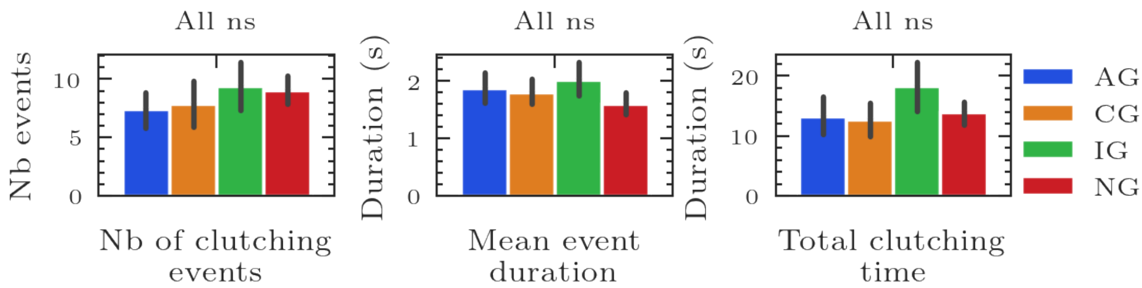


Figure 7.7: Collected data about mapping clutching under the different conditions (N=12).

In order to provide a preliminary evaluation of the effect of clutching on performance, the execution errors during a short period of time (2 s) before and after each mapping recalibration are computed for the twelve participants of the previous user study. For each event, the mean execution error norm $||\epsilon_h||$ is computed for the 2 s preceding and following the event. Then, the difference between the pre and post clutching can be computed for each event.

Once averaged across subject, the overall effect of clutching can be studied for each condition. Under **CG**, the execution error is on average 10% higher after the clutch events than before (see Figure 7.8). Similarly, there is a 9.8% increase under **AG**, but neither increase is statistically significant (p $>$ 0.1). However, clutch events have a significant impact during **IG** and **NG** scenarios and locally increase the execution error by 16% ($p < 0.01$) and 26% ($p < 0.05$) respectively.

The clutch events did not have a statistically significant impact on the task execution accuracy when the guidance is correct. The fact that a significant error is introduced under **IG** at clutch events could be explained by the larger guidance forces applied

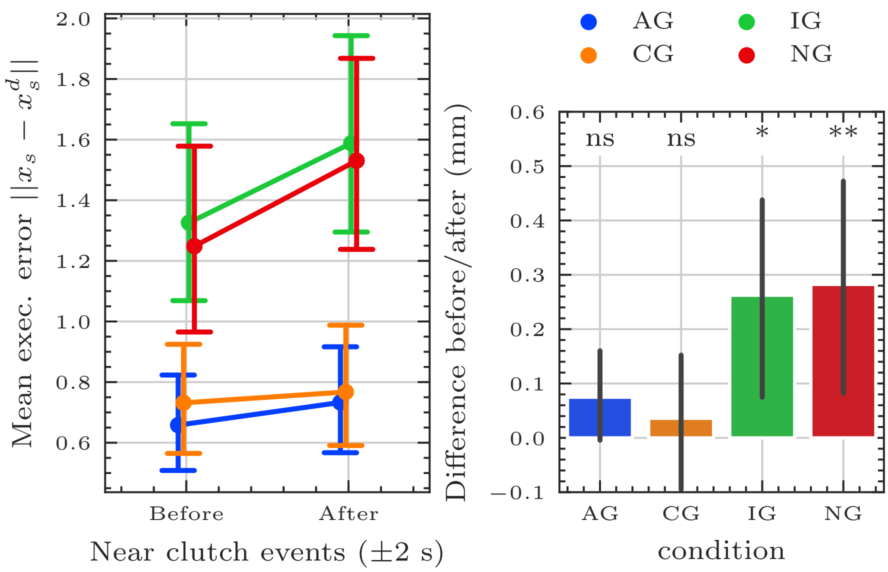


Figure 7.8: Comparison of the execution mean errors over the 2 s time periods before and after the clutch events. Significance levels ns ($p > 0.05$), $*$ ($p < 0.05$), $**$ ($p < 0.01$) for the differences were computed using a Wilcoxon Sign test.

during the recalibration procedure. But the same phenomenon also appears when haptic guidance is not provided (**NG**), suggesting that the force applied during clutching may not be the only source of errors. A possibility is that under **NG**, the participants need some time to readjust after the clutch events in order to regain a better sense of how their actions affect the robot. Also, only minimal damping is introduced under **NG**, but damping has a stabilizing effect that might help the participants to cope with seemingly destabilizing effects introduced by the clutching process. This might partially explain the fact that participants were less affected under **AG** and **CG**. Then, although the effects of clutching could not be conclusively quantified from this experiment, it provides us with interesting insights for future developments.

7.5 Conclusions

We presented a comprehensive study with participants that provides results about the effects of both inaccurate and adaptive haptic guidance. Although the main objective is to evaluate the adaptive guidance approach introduced in the previous chapters, the results of this user study are more generally relevant to teleoperation with haptic guidance. From the experimental results, we can conclude that when task and/or robot models are inaccurate, the performance is negatively impacted: the accuracy is reduced, the movements are less smooth, and the user experience is degraded. In this scenario, adaptive guidance was then found to be preferable to non-adaptive guidance, both objectively and subjectively.

Discussion

In the following, we provide a synthetic discussion about the effect of inaccuracies in guidance as well as conclusions drawn from the user study and the other experiments with an operator in-the-loop. Additionally, an alternative learning method complementary to the fading memory EKF is presented, allowing us to compare the advantages and drawbacks of both. Finally, we briefly discuss the passivity of the generated adaptive guidance.

To guide or not to guide?

The intuition that motivated the work presented in Part II is that a learning-based adaptive guidance would significantly mitigate the effects of registration inaccuracies. We found this to be the case for the tasks considered in our experiments: not only the performance was objectively improved (i.e., accuracy and smoothness), but the qualitative evaluation by participants was overall very positive.

In all the scenarios we considered, the parameters could be learned from teleoperation data only, but this is not necessarily always the case. As is discussed afterwards, including other sources of information (e.g., image measurements) would be a possibility to cope with identifiability issues. Nonetheless, it is also pertinent to reflect on the best policy to follow when parameters cannot, in fact, be corrected. The results from Chapter 7 suggest that, in such a case, it might be preferable not to provide haptic guidance, eventually to reintroduce the guidance once the models have been improved. However, even inaccurate guidance was found to be beneficial in some telerobotic scenarios (Oosterhout et al. 2015), such that the strategy to adopt will likely depend on the nature of the task.

Something that could be interesting, rather than not guiding the operator at all, would be to reduce the guidance authority (i.e., reduce the stiffness) until it becomes possible to update the parameters. Ideally, the stiffness would reflect the uncertainty over parameter estimates as proposed by Zeestraten et al. (2018). However, although Kalman filtering usually makes it possible to obtain an estimate of this uncertainty, the fading memory factor introduced in Chapter 5 is known to affect the estimation covariance P_k in such a way that it has little physical meaning and is not, in fact, equivalent to the real parameter estimates covariance (Simon 2010, Chapter 7). Therefore, the covariance matrix of the fading memory cannot be used as it is to tune the stiffness of the guidance online. Further work will then be necessary to implement a variable stiffness guidance.

It is also worth discussing the guidance along the tangent direction to the guidance reference trajectory. We found that it helps smoothing the movements of the operator by filtering the task execution velocity imposed by the operator, which in turn makes learning the time parameters easier. Nonetheless, the study with participants we carried out revealed that the participants did not unanimously appreciate the guidance along the trajectory due to perceived damping. We could reduce the phenomenon by actively pulling the operator as in (Papageorgiou et al. 2020) or simply not guiding along the tangent to the trajectory. Actually, it should be noted that guiding the operator along the trajectory is not that common in telerobotics and, usually, virtual fixture-like guidance is preferred. It would then be interesting to evaluate how different haptic guidance strategies impact the performance for tasks of varied curvature and complexity through a dedicated user study.

Sliding window-based learning – an alternative to EKF

When introducing the fading memory EKF filter approach for simultaneous task and robot model parameters learning in Chapter 5 (see Section 5.3), we did so by approximating the solution of a more generic learning problem defined in Equation (5.12). As already commented, the proposed fading memory EKF is therefore not the only possible implementation. Notably, another interesting approach consists in learning the model parameters using a sliding window of collected measurements such that more information is used for learning. However, when using a window of observations, the time parameterization can potentially differ between the observations (e.g., over a part of the window, the velocity is null because the operator paused). We therefore proposed an approximated solution to learn the time parameters independently from the other model parameters. The method is briefly presented in the following section and is detailed in Appendix E. In the following, we discuss the implications of a window-based learning compared to the fading memory EKF presented in the Part II of this thesis and conclude with perspectives concerning the implementation of the learning.

An adaptive size sliding window learning method for online task registration

In (Poignonec et al. 2021), we proposed a learning-based on an optimization performed over a sliding window of sampled robot positions. Note that the formulated learning problem and resulting method assume therefore that the robot model parameters are known or that Cartesian robot pose estimation is available. The proposed approach is akin to a non-rigid path registration such that the geometry and the dynamics of the task become (nearly) independent problems:

- the parameters encoding the geometry of a task (i.e., all but the time parameters) are learned by minimizing the distance between the sampled follower robot position (resulting from the teleoperation by an operator) and the path underlying the estimated desired trajectory as depicted in Figure 7.9a. This optimization is performed over a sliding window and the cost computed as the sum of the squared distances between the sampled follower robot positions and their closest point on the desired path (see Figure 7.9b). Furthermore, an algorithm is devised to adjust the size of the sliding window online. Details can be found in appendix sections E.2.1 and E.2.2;

- the time parameters are estimated separately, using only the most recent follower robot position measurement such that the estimated desired position tracks the follower robot motion (see Figure 7.9c). The haptic guidance is then generated from the estimated desired follower robot trajectory obtained from the geometrical parameters estimated over the sliding window and the local tracking of the time parameters. Details can be found in appendix Section E.2.3.

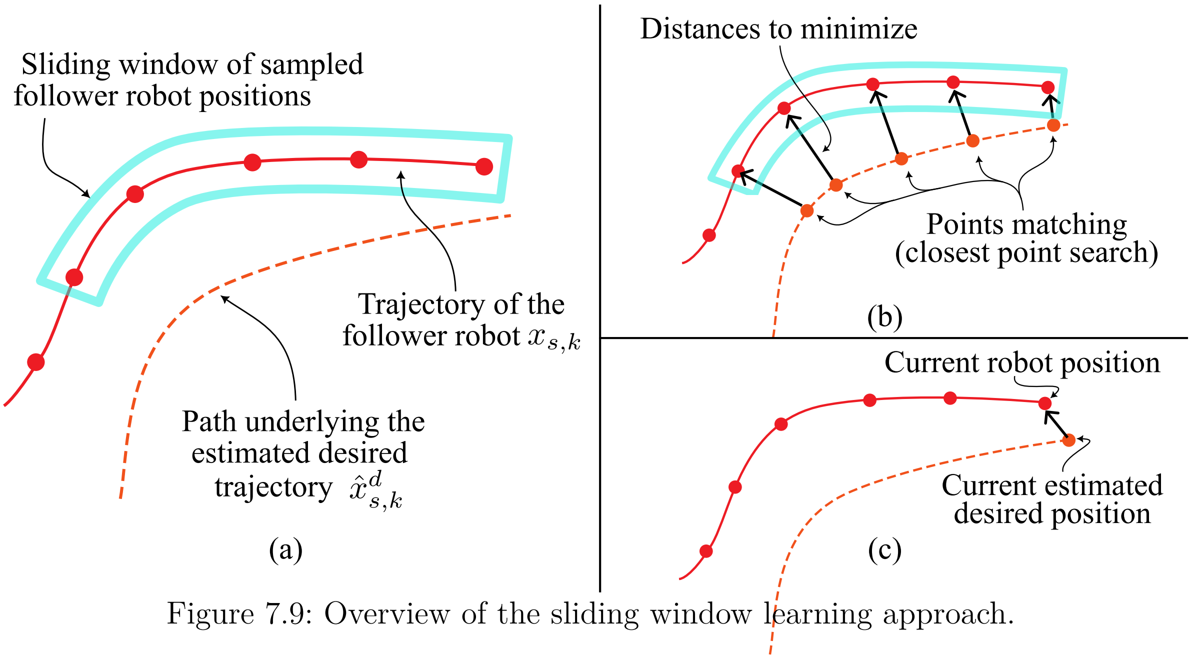


Figure 7.9: Overview of the sliding window learning approach.

Comparison of EKF with sliding window-based learning

The fading memory EKF and the sliding window approaches have noticeably different properties. The most obvious difference is that the EKF is memoryless, whereas the sliding window approach uses data sampled over a possibly large time horizon. This significantly impacts execution time, as the sliding window approach involves the inversion of large matrices and also relies on numerous line search optimizations to compute the closest point matchings for each sampled robot position. Nonetheless, note the update rate we could achieve with the sliding window approach was sufficient for learning as demonstrated by the experimental results.

Another major difference lies in the way the time parameterization is modeled and subsequently learned. The fading memory EKF learns the time parameters along the others whereas the sliding window approach relied on a pure geometrical approach, simply considering ψ as a free variable. Ignoring the time parameterization in such a way allows to reduce the problem complexity, but it ignores the dynamic aspect of the task when learning. In (Poignonec et al. 2021), we presented experimental results with a high curvature desired trajectory (sinusoidal curve) to illustrate an identified limit of our sliding window approach: line search local minima. As illustrated in Figure 7.10, some scenarios can lead the line search (i.e., search of the closest points from robot position $x_{s,k}$ on the path underlying $\hat{x}^d_s$) to converge towards inaccurate values when there are large registration errors, typically near high curvature sections. Consequently, the learning might not converge towards the correct parameters since the evaluated loss function will have an artificially small value. Although the fading memory EKF could also be impacted in such scenarios, it was found to be less subject to these issues due to the smoothing of $\hat{\psi}_k$ induced by the time parameters tracking.

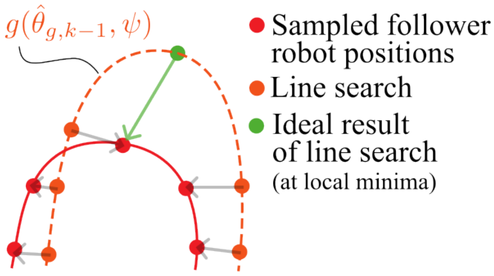


Figure 7.10: Illustration of the effect of line search (i.e., closest point search) local minima on the point matching procedure (see Equation E.6).

So far, solely the drawbacks of the sliding window approach were discussed, but it also has one major advantage compared to memoryless learning implementations such as the fading memory EKF. The fact that the learning uses a window of observations means that the stored data can also be used to test parameters identifiability online, thus considering the data actually available for learning as opposed to relying on offline computation based on assumptions (see Section 6.2). Actually, the criterion used for window size management (see γ_k in appendix Section E.2.2) could be defined as an identifiability criterion similar to the one used in (Self et al. 2019), where the authors devise a metric based on the singular values of the regressor to decide whether the parameters should be updated or not. This type of scheme would allow the window to expand until the parameters become identifiable.

A perspective of this work would then consist in devising a learning method that combines the advantages of both filtering and sliding window approaches. For instance, moving horizon estimation, i.e., MHE (Rawlings 2015), could be an interesting compromise. Although MHE is by definition a sliding window learning method, it also makes use of a Bayesian formulation similar to the EKF (actually, MHE is equivalent to EKF for a window size of one). It would be more computationally demanding that the EKF, but previous studies demonstrated that MHE could be implemented efficiently and that for small window size, the achievable update rate was comparable (Kühl et al. 2011). MHE would then combine advantages from the EKF (e.g., time parameters and noise modeling) and from the sliding window method (i.e., the possibility to run tests on the collected data). Furthermore, it would allow the integration of constraints to define the admissible set of parameter values.

Passivity of adaptive haptic guidance

Except if the haptic guidance is turned on and off as discussed in Section 7.4 (i.e., during clutching), the guidance implementation with constant stiffness and critically damped behavior is already passive under the commonly used definition (Hannaford et al. 2002; Kronander et al. 2016). Furthermore, the update of model parameters has the effect of reducing the guidance errors since the cost function minimized by the learning process is, *in-fine*, the guidance error[2]. Therefore, the learning itself does not increase the energy stored into the virtual mass-spring-damper system, but the time parameterization can, occasionally, lead to active behaviors.

The haptic guidance as experienced by the operator is in fact not strictly passive since the haptic guidance potentially injects energy when the estimated desired velocity is not null (i.e., $\dot{\psi}_k \neq 0$). This is typically the case if the operator suddenly stops: the time parameters will take some time to be corrected and the guidance reference will keep going forth. Papageorgiou et al. (2020) used a passivity tank to avoid active behaviors when providing a look-ahead haptic guidance such that the guidance reference is computed for a future time (e.g., 0.1 s in the future) and therefore pulls the operator along the desired trajectory. The energy tanks-based approach proposed in (Papageorgiou et al. 2020) might be a good solution to enforce the passivity in our case as well. To further minimize the disturbance of operator gestures, a constraint could also be added on the rate of energy transfer in a similar way to the so-called energy-aware control schemes, where the energy generated and stored by the robot (resulting from inertia, typically) is monitored and limited to implement safety features for physical human-robot interaction (Raiola et al. 2018; Meguenani et al. 2016).

[2] In the case of the fading memory EKF. This is not necessarily true if a window of observations is used since the cost would be computed from past observations, thus not the current guidance error.

Part III

Final conclusion and perspectives

Conclusion and perspectives

Conclusion

In Part I, we proposed several complementary methods for the *in-situ* backlash model identification of cable-actuated flexible endoscopes. Compared to existing *in-situ* backlash model identification methods, the proposed approaches allow the use of more complex backlash model, which is pertinent to real-world scenarios. All the proposed methods are very generic and can be applied to different robot architectures and different intra-operative sensors. Notably, the original problem formulation in Chapter 3 allows the online backlash width identification of endoscopes with eye-in-hand camera configurations, which was not possible using previous identification methods. Furthermore, the proposed methods were evaluated in simulation and on real systems: a clinical colonoscope in Chapter 3 and the cable-actuated endoscopic tool of a endoluminal surgical platform in Chapter 4. These experimental results showed that the methods could be used to retrieve the correct backlash width model from endoscopic images only.

In Part II, we proposed novel adaptive guidance schemes adapted for telerobotics. The main contribution is the development of an online task and robot kinematic models parameters learning method that uses the operator in-the-loop as a source of information. The method is applicable when the models are known, but that the values of some underlying parameters are not. This is typically the case when a task has been defined but parameters have to be adapted to the actual environment during its execution. Different challenges were identified and solutions put forth. Notably, a tracking of the desired trajectory's time parameterization was proposed to cope with variable velocity imposed by the operator, which is critical in a teleoperation context. The proposed implementation of the learning problem was evaluated through several comprehensive experiments, both to characterize the learning performance objectively and, more qualitatively, through a study with participants. The experimental results support the idea that an adaptive guidance is preferable to its non-adaptative counterpart when models are initially inaccurate, which would support its use during robot-assisted MIS where registration errors are frequent. Such methods are a good compromise to use automation based on a pre-operative plan while leaving the final control to the surgeon, thus bridging the gap between manual and automatic task execution in surgical robotics. As Yoda once said, although not of haptic guidance, "For my ally is the Force, and a powerful ally it is" (*The Empire Strikes Back*, 1980). However, for the force to truly be an ally in the realization of teleoperated robotic tasks, we argue that it should also learn from its human partner throughout the realization of the task.

Much work is still needed until such methods can reach an operating room. Nonetheless, we hope that our work will contribute to the development of more flexible and adaptable assistive systems that will, in the future, assist during minimally invasive surgery.

Perspectives and future work

In addition to the more technical and focused perspectives presented at the end of parts I and II, we give here other avenues for future research.

Improving the online learning of model parameters

The stability-plasticity dilemma

In this thesis, we mostly considered stationary model parameters for experimental validation such that, although the parameters are initially incorrect, the correct values do not change over time. However, parameters might vary online and, in the context of surgical surgery, they would most likely do. For instance, the task registration could be invalidated by tissue motion or the pose of a surgical instrument held by a gripper could change due to slippage. Similarly, backlash model parameters necessary for the control of an endoscope might vary *in-situ* due to a change in the shape of the (passive) flexible body of the endoscope.

Experimental results showed that the fading memory EKF could cope with a sudden (small) change in correct parameters (see Section 6.3.3), but it requires a relatively small fading factor α. Similarly, a learning method based on a sliding window as mentioned in the discussion of Part II could cope with such a change, but only if the window is small enough. However, tuning a learning method so that it can cope with non-stationary correct parameters would also reduce the robustness, since only the latest observations would be given a significant weight. This trade-off between the ability to cope with change and the robustness of the learning is sometimes referred to as the *stability-plasticity dilemma* (Gama et al. 2014).

An interesting complement to our work would consist in detecting such changes to implement different strategies: if the correct parameters are stationary, then robustness is maximized; if the correct parameters are changing or have changed, the tracking capabilities are maximized. For instance, if a change in the correct parameters is detected, one could temporarily decrease the fading factor in the methods proposed in chapters 2, 3, and 5. Then, once the estimation errors have been reduced, the fading factor would be returned to its nominal value (see Figure 7.11b). Similarly, if the learning is performed using a window of observation, a straightforward approach would consist in refreshing the content of the window: when a change of parameters occurs, the size of the window is reduced, but then is filled back up with the latest observations (see Figure 7.11c).

Different solutions for parameters change detection have been proposed in the literature (Gama et al. 2014). For instance, an approach that would be applicable to both sliding window-based and EKF-based learning consists in monitoring the residuals of the learning process (i.e., remaining prediction errors after parameters update). The so-called cumulative sum (CuSUM) metric is obtained by applying a negative offset

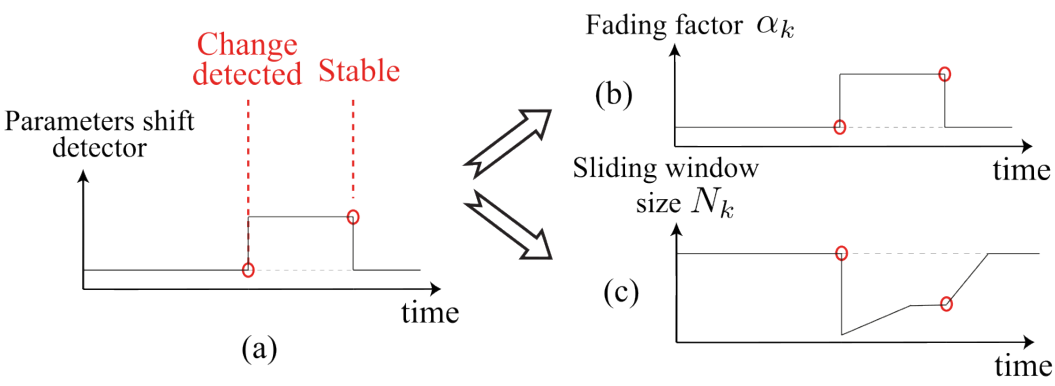


Figure 7.11: (a) Parameters change detection could be used to modify the properties of the learning process temporarily. (b) For a fading memory filter. (c) For a sliding window-based learning process.

to the learning residual (i.e., threshold) and to compute the cumulative sum of the resulting number (Page 1954). If the CuSUM is above a predefined threshold, the parameters are assumed to have changed. We have conducted some preliminary tests on the basis of this concept to vary the fading memory factor when the task parameters suddenly change in a context similar to the one from Section 6.3.3. It seems to be a promising way to achieve both tracking and steady state estimation accuracy. In future work we might then investigate how to implement this on a medically relevant scenario.

Multi-modal learning

The learning method proposed in Chapter 5 was designed to also accommodate external measurements, but the main line of investigation remained the correction of haptic guidance from information extracted from the operator in-the-loop. Consequently, the fusion of teleoperation data and exteroceptive measurements was not investigated beyond some preliminary simulation studies.

However, the possibility of including additional measurements is a very interesting feature of the proposed algorithm, since such measurements could not only improve learning, but also make it possible in scenarios where human actions alone do not yield rich enough information. For instance, consider a scenario where partial visual measurements (i.e., of a part of the robot or the task) are available, but where they are not sufficient to learn either the task or robot model. Such measurement could still augment those extracted from the presence of the operator, hence increasing achievable accuracy. Furthermore, scenarios could arise where the models parameters are unidentifiable if only the presence of the operator is used for learning, but become identifiable once more sources of information are used. Actually, such measurements do not have to be acquired with an endoscopic camera; potential sensory inputs also include force sensing if present and any real-time medical imaging modality.

Conversely, in chapters 2 and 3, only measurement acquired from exteroceptive sensors were considered, but the online backlash estimation could potentially be improved by also extracting information from the operator in-the-loop when present. However, doing so would require knowledge about the intention of the operator and, therefore, would likely lead to the need for online task model estimation. Such simultaneous backlash and task models learning is a natural perspective of our work, which we further discuss in the following section.

Simultaneous backlash and task models correction

We proposed online backlash estimation methods for robotized cable-actuated endoscopes in the first part of this thesis and focused on task registration and robot modeling issues throughout the second part. However, combinations of the two problems is also of great interest for medical applications; for instance, the realization of an inaccurately registered task using an endoscope whose backlash modeling is likewise inaccurate. Let us consider a scenario with an eye-in-hand endoscope (see Chapter 3) where a contactless task has to be performed. The realization of the task by a surgeon could be used to extract information about both the backlash and the registration of the task such that models can be corrected.

The learning methods we used for backlash estimation (i.e., fading memory DEKF in chapters 2 and 3) and for task parameters correction from using inputs (i.e., fading memory EKF in Chapter 5) are compatible and can be used together. However, some practical challenges remain. For one, the effect of (uncompensated) backlash on haptic guidance will have to be investigated. Contrarily to the continuous errors that arose in Part II, the guidance inaccuracies resulting from inaccurate backlash modelling will also depend on the history of motor displacements. Furthermore, assessing parameters identifiability will be more delicate due to the discontinuous backlash model. Nonetheless, schemes of this type would be an interesting perspective building on different aspects of our work to provide a novel way to assist during the teleoperation of endoscopes.

Towards more autonomy

The work on endoscope backlash identification presented in Part I is readily applicable to automatic task execution since only intra-operative measurements (i.e., endoscopic images) are used for learning. A potential use case would consist in an automatic backlash model identification stage followed by an automatic task realization. Nonetheless, practical challenges such as task registration would remain and implementing an automatic mode might sometimes be difficult. In such scenarios, the method for online task and robot models parameters learning we presented in Part II could be used to correct the models parameters from a surgeon performing the task through teleoperation. Once the parameters are corrected, the system could then complete the realization of the task, still under supervision, but automatically.

Appendix A

Detailled list of scientific communications

The work presented in this thesis has been the object of the following national and international scientific communications:

International peer-reviewed journal publications

RA-L – 2020 Work presented in Chapter 4 and published as (Poignonec et al. 2020):
T. Poignonec, P. Zanne, B. Rosa, and F. Nageotte, "Towards In Situ Backlash Estimation of Continuum Robots Using an Endoscopic Camera," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4788–4795, Jul. 2020.

Under review Work presented in chapters 5, 6, and (partially) 7:
T. Poignonec, F. Nageotte, N. Zemiti, and B. Bayle, "Online correction of task registration and robot models from user input."

Under review Work presented in Chapter 3:
T. Poignonec, P. Zanne, N. Zemiti, B. Bayle, and F. Nageotte, "Image-based backlash width identification for eye-in-hand robotized flexible endoscopes."

International peer-reviewed conferences

ACC – 2023 Work presented in Chapter 2:
T. Poignonec, F. Nageotte, and B. Bayle, "A fading memory discontinuous EKF for the online model identification of cable-driven robots with backlash," in *IEEE American Control Conference* (ACC), June 2023.

ICRA – 2021 Work discussed in the closing chapter of Part II (detailed in Appendix E) and published as (Poignonec et al. 2021):
T. Poignonec, F. Nageotte, N. Zemiti, and B. Bayle, "Simultaneous haptic guidance and learning of task parameters during robotic teleoperation – a geometrical approach," in *IEEE International Conference on Robotics and Automation* (ICRA), May 2021, pp. 3619–3625.

IROS – 2020 Presentation of the work published in (Poignonec et al. 2020).

National conferences

CAMI DAYS – 2022 Presentation to the annual scientific seminar of the CAMI LabEx[1]. Title of the presentation: "In-situ and online backlash identification for tendon-driven flexible robots." Montpellier, November 25th, 2022.

CAMI DAYS – 2021 Presentation to the annual scientific seminar of the CAMI LabEx[1]. Title of the presentation: "Simultaneous task and robot models learning for teleoperation." Paris, November 11th, 2021.

ICube scientific seminar – 2020 Presentation to the annual scientific seminar of the Robotic Data and Healthcare (RDH) team. Title of the presentation: "How to guide and learn simultaneously? New haptic shared-control schemes for surgical robotics." Strasbourg, July 3rd, 2020.

Scientific mediation

PapierMaché – 2021 Article titled "Des robots aux gants de velours – Une introduction au contrôle en impédance," papiermachesciences.org[2].

[1] Computer Assisted Medical Interventions (CAMI) LabEx, **ANR-11-LABX-0004-01**

[2] Full url: https://papiermachesciences.org/2021/07/10/des-robots-aux-gants-de-velours-une-introduction-au-controle-en-impedance/?v=A (visited on 03/14/2023).

Appendix B

Kinematic model of the considered cable-actuated endoscopes

B.1 Eye-to-hand 3 DOF endoscope (chapters 2 and 4)

As detailed in (Nageotte et al. 2020), the tip position of the 3 DOF (translation, bending, and orientation) flexible robot depicted in Figure 2.11 can be computed as

$$x_k = \mathcal{K}(c_k) = \begin{cases} \begin{bmatrix} \left(\dfrac{L}{c_{2,k}}(1-\cos(c_{2,k})) + d\sin(c_{2,k})\right)\cos(c_{3,k}) \\ \left(\dfrac{L}{c_{2,k}}(1-\cos(c_{2,k})) + d\sin(c_{2,k})\right)\sin(c_{3,k}) \\ c_{1,k} + \dfrac{L}{c_{2,k}}\sin(c_{2,k}) + d\cos(c_{2,k}) \end{bmatrix}, & \text{if } c_{2,k} \neq 0 \\ \begin{bmatrix} 0 \\ 0 \\ c_{1,k} + L + d \end{bmatrix}, & \text{if } c_{2,k} = 0 \end{cases} \tag{B.1}$$

B.2 Eye-in-hand 2 DOF endoscope (Chapter 3)

Mapping from cable displacement into configuration variable

The mapping from (distal) cable displacement $c_k \in \mathbb{R}^2$ into configuration variables $\phi_k \in \mathbb{R}$ and $\beta_k \in \mathbb{R}$ is

$$\begin{bmatrix} \phi_k \\ \beta_k \end{bmatrix} = \mathcal{K}_c(c_k) \tag{B.2}$$

where

$$\phi_k = \begin{cases} 0, & \text{if } c_{1,k} = c_{2,k} = 0 \\ \operatorname{atan2}(-c_{2,k}, -c_{1,k}), & \text{otherwise} \end{cases} \tag{B.3}$$

and

$$\beta_k = \frac{2}{D}\sqrt{c_{1,k}^2 + c_{2,k}^2} \tag{B.4}$$

Finally, D is the distance between opposed cable fixtures on the distal side of the bending section as illustrated in Figure 3.2.

Mapping configuration variable to Cartesian pose

The Cartesian pose of the robot is computed as

$${}^{b}T_c(c_k) = \begin{bmatrix} {}^{b}R_c(c_k) & {}^{b}t_c(c_k) \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} = \mathcal{K}_p(\phi_k, \beta_k) \tag{B.5}$$

where ${}^{b}t_c(c_k) \in \mathbb{R}^3$ is the Cartesian position of the camera in the frame $\mathcal{F}_b$ (see Figure 3.2, right) and ${}^{b}R_c(c_k) \in SO(3)$ is the orientation matrix of the camera frame $\mathcal{F}_c$ w.r.t. $\mathcal{F}_b$. As detailed by Ott et al. (2011), the Cartesian pose of the robot is computed as

$${}^{b}t_c(c_k) = \begin{cases} \begin{bmatrix} \left(\dfrac{L}{\beta_k}(1-\cos(\beta_k)) + d\sin(\beta_k)\right)\cos(\phi_k) \\ \left(\dfrac{L}{\beta_k}(1-\cos(\beta_k)) + d\sin(\beta_k)\right)\sin(\phi_k) \\ \dfrac{L}{\beta_k}\sin(\beta_k) + d\cos(\beta_k) \end{bmatrix} & \text{, if } \beta_k \neq 0 \\ \begin{bmatrix} 0 \\ 0 \\ L+d \end{bmatrix} & \text{, if } \beta_k = 0 \end{cases} \tag{B.6}$$

and

$${}^{b}R_c(c_k) = \begin{bmatrix} \mathrm{s}^2\phi_k + \mathrm{c}\beta_k\mathrm{c}^2\phi_k & -\mathrm{s}\phi_k\mathrm{c}\phi_k(1-\mathrm{c}\beta_k) & \mathrm{c}\phi_k\mathrm{s}\beta_k \\ -\mathrm{s}\phi_k\mathrm{c}\phi_k(1-\mathrm{c}\beta_k) & \mathrm{c}^2\phi_k + \mathrm{c}\beta_k\mathrm{s}^2\phi_k & \mathrm{s}\phi_k\mathrm{s}\beta_k \\ -\mathrm{c}\phi_k\mathrm{s}\beta_k & -\mathrm{s}\phi_k\mathrm{s}\beta_k & \mathrm{c}\beta_k \end{bmatrix}$$

where ϕ_k and β_k are computed from c_k using Equation (B.2).

Appendix C

Identifiability analysis of the task and robot parameters (Chapter 6)

For this analysis, continuous time will be used such that time varying variables are denoted with (t). In order to build the dynamical system equivalent to the task and robot parameters learning, the explicit time dependency is removed by modeling ψ as the time-integral of $\dot{\psi} = b$ with b the parameter encoding the velocity. The task and robot parameters as previously defined (i.e., $\theta = [\theta_g\ \theta_r]^T$) are then split between time-varying state variables $\psi(t)$ and stationary parameters θ (i.e., parameter a removed from the definition of θ in Chapter 6) such that

$$\theta = \begin{bmatrix} b & r_z & t_x & t_y & L \end{bmatrix}^T \tag{C.1}$$

The parameters can be included in the state of the system by considering a state transition model $\dot{\theta}(t) = 0$. We would then get an equivalent system with dynamic state $\mathcal{X}(t) = [\psi(t)\ \theta^T(t)]^T$, known input $u(t) = q_s(t)$, and measurement $z(t) = h\left(\mathcal{X}(t), u(t)\right) = \epsilon_h(t)$, where $\epsilon_h(t) = x_s^d(t) - x_s(t)$. However, with the chosen experimental setup, the orientation of the tool is fixed such that we artificially choose the end-effector position $u(t) = x_s^{ee}(t)$ as the input of our system, in practice computed from $q_s(t)$. The robot model is then rewritten such that the tool tip position $x_s(t)$ is computed from $x_s^{ee}(t)$ and the fixed end-effector orientation ${}^wR_{ee}$. Finally, we obtain the dynamic system:

$$\begin{cases} \mathcal{X}(t) = \begin{bmatrix} \psi(t) \\ \theta(t) \end{bmatrix} & \text{(C.2)} \\ \dot{\mathcal{X}}(t) = f\left(\mathcal{X}(t), u(t)\right) & \text{(C.3)} \\ z(t) = h\left(\mathcal{X}(t), u(t)\right) & \text{(C.4)} \end{cases}$$

where $f(\cdot)$ is the state transition model such that

$$f\left(\mathcal{X}(t), u(t)\right) = \begin{bmatrix} \dot{\psi}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

and $h(\cdot)$ is the observation model defined as

$$h\left(\mathcal{X}(t), u(t)\right) = \left(\begin{bmatrix} t_x \\ t_y \\ 0 \end{bmatrix} + {}^wR_t(r_z)\Gamma(\psi(t)) \right) - \left(x_s^{ee}(t) + {}^wR_{ee} \begin{bmatrix} 0 \\ 0 \\ L \end{bmatrix} \right) \tag{C.5}$$

We can then analyze the observability of this system. An observable system is defined as a system whose initial state $\mathcal{X}(0)$ can be correctly estimated from the observation of the input $u(t)$, the output $z(t)$, and their respective derivatives. The observability analysis consists in building the observability matrix $\mathcal{O}(\mathcal{X})$ of the system system and to verify that its rank is full. It should be noted that due to the non linearity of the system, the observability matrix is built using Lie algebra and, more precisely, the so-called extended Lie derivatives of the system (Karlsson et al. 2012). To do so, we use the *STRIKE GOLDD toolbox*[1]developed by Villaverde et al. (2019), a Matlab script using the symbolic toolbox to compute $\mathcal{O}(\mathcal{X})$ and its rank symbolically under different assumptions about the type of inputs (Villaverde et al. 2019; Martínez et al. 2020). If the matrix is full rank, the augmented system is structurally observable and by extension, the parameters θ are structurally identifiable and $\psi(t)$ is observable. It should be noted that, by construction, the system is equivalent to the initial parameters learning problem. Therefore, the results of the analysis performed on this system is also valid for the initial learning problem.

As the observability heavily depends on the implemented task model, especially the underlying path $\Gamma(\psi)$ that is then registered according to the task parameters, the analysis is performed on a generic path of constant curvature. This simplified path model can locally approximate other paths such that the structural identifiability analysis of the parameters can then be generalized. We therefore distinguish three scenarios as follows.

No motion

Unsurprisingly, if the operator does not move such that $b = 0$, $\forall t > 0$, then the task and robot model parameters are found to be structurally unidentifiable. This can be seen as an excitation condition: if the operator takes no action, then no information can be extracted.

The path is a straight line

When path $\Gamma(\psi)$ underlying the task model is a straight line on the drawing surface, the analysis of the observability matrix reveals that only the parameters encoding the task velocity (i.e., b) and the orientation of the path w.r.t. the world frame of reference (i.e., r_z) are identifiable. This is due to the fact that both $\psi(0)$ and a combination of t_x, t_y, and L have the same effect on the resulting task, namely translating the line along its tangent. The redundancy of these parameters causes the lack of observability and identifiability, respectively.

The path is an arc of constant curvature

Let Γ_Π be a path expressed in the drawing frame of reference such that

$$\Gamma(\psi) = {}^tR_\Pi \Gamma_\Pi(\psi) + {}^tO_\Pi \tag{C.6}$$

[1]Source code available at https://github.com/afvillaverde/strike-goldd

where ${}^{t}R_{\Pi}$ and ${}^{t}O_{\Pi}$ are, respectively, the rotation matrix and translation from the task frame attached to the base of the tilted support to the drawing frame (see experimental setup). The in-plane path is then defined as

$$\Gamma_{\Pi}(\psi) = r \begin{bmatrix} \cos(\frac{1}{r}\psi) - 1 \\ \sin(\frac{1}{r}\psi) \\ 0 \end{bmatrix} \tag{C.7}$$

where r is the radius of curvature of the arc. The different desired trajectories $x_s^d(t)$ are then generated by a variation of r (see Figure C.1, left).

The rank of the observability matrix was full for generic initial conditions (rank computed symbolically and with random numerical values) using the extended Lie-derivative up to the first order. The task and robot model parameters were then found to be structurally identifiable and the state variable $\psi(t)$ observable, except when the radius of curvature is infinitely large (i.e., the desired trajectory is a straight line). However, although the task and robot parameters were found to be (locally) fully identifiable anywhere the curvature of the task is finite, this is only a theoretical result based on the rank of the observability matrix $\mathcal{O}(\mathcal{X})$. In practice, the degree to which a system is observable can greatly vary and this relative observability can be quantified. Let the observability metric be the conditioning of the observability matrix computed as

$$cond(\mathcal{O}) = \frac{\lambda_{max}(\mathcal{O})}{\lambda_{min}(\mathcal{O})} \tag{C.8}$$

where $\lambda_{max}(\mathcal{O})$ and $\lambda_{min}(\mathcal{O})$ are, respectively, the largest and smallest singular values of the matrix $\mathcal{O}(\mathcal{X})$. The lower $cond(\mathcal{O})$ is, the more observable the system is as long as $\lambda_{min}(\mathcal{O}) > 0$.

The effect of curvature on the local observability is illustrated in Figure C.1 where a range of radiuses of curvature (i.e., $r \in [1.10^{-2}; 3.10^{-1}]$ m) leads to a better and rather constant observability metric conditioning. For radiuses out of this range, the observability decreases. The decrease of observability for large values of r is explained by the analysis provided previously with a line as underlying path of the desired trajectory. This local analysis shows that the dynamic system equivalent to the learning problem is fully observable anywhere the curvature is finite (i.e., not a straight line). However, the observability decreases as the curvature of the task leaves the approximative range $r \in [10\text{ mm}; 300\text{ mm}]$ and then, so would the structural identifiability of the models parameters θ.

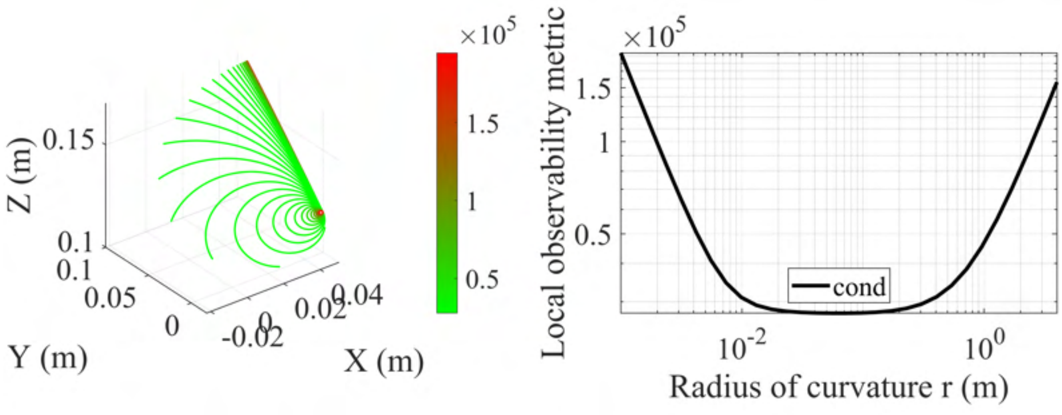


Figure C.1: Observability metric $cond(\mathcal{O}(\mathcal{X}))$ computed for different radii of curvature of the task r, a task execution velocity $\dot{\psi}(t) = 0.01\text{m.s}^{-1}$, and an arbitrary set of task and robot model parameters. A lower value of $cond(\mathcal{O}(\mathcal{X}))$ is desirable. Left: trajectories $x_s^d(t)$ colored according to the resulting value of $cond(\mathcal{O}(\mathcal{X}))$. Right: detailed values on a log-log scale.

Appendix D

User study (Chapter 7)

D.1 Complement to Figure 7.2

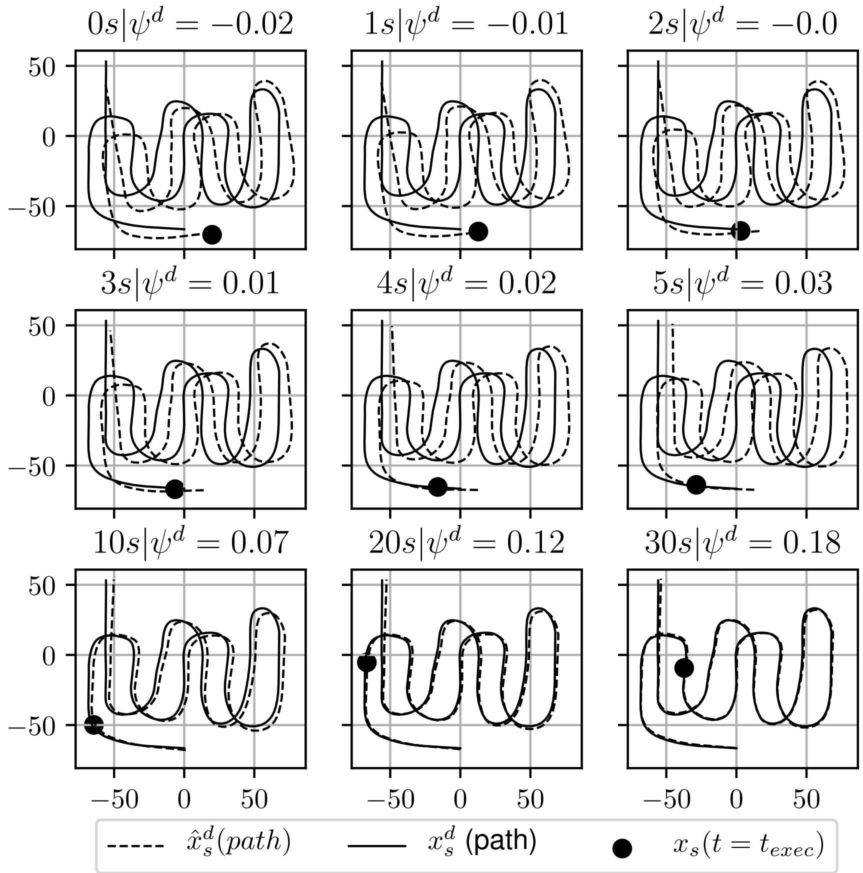


Figure D.1: Correct and estimated trajectories x_s^d displayed in the XY plane of the drawing frame under condition **AG**. For each snapshot, the execution time (not including *clutching* time) and the task progress is provided.

D.2 Questionnaire for subjective performance assessment

ID du participant : ________________ **ID de la tâche :** ____________________ **Date :** ________

Merci de remplir les points ci-dessous pour la dernière tâche que vous avez réalisée.

Pour chaque affirmation, cochez la case qui représente le mieux votre ressenti :

1 – Je me sentais en contrôle lors de l'exécution de la tâche

☐ Pas du tout d'accord ☐ Plutôt pas d'accord ☐ Je ne sais pas ☐ Plutôt d'accord ☐ Tout à fait d'accord

2 – Les forces générées par l'interface maître m'ont aidé à réaliser la tâche

☐ Pas du tout d'accord ☐ Plutôt pas d'accord ☐ Je ne sais pas ☐ Plutôt d'accord ☐ Tout à fait d'accord

---

3 - Indice de charge de travail de la NASA

La méthode de l'indice de charge de travail de la NASA de Hart et Staveland évalue la charge de travail sur cinq échelles à sept points. Des incréments correspondant à des estimations élevées, moyennes et faibles pour chaque point donnent lieu à 21 gradations sur les échelles.
Remarque : La performance va de « bonne » à gauche vers « mauvaise » à droite.

Demande mentale : Comment était la demande mentale de cette tâche ?

Très faible — Très élevée

Demande physique : Comment était la demande physique de cette tâche ?

Très faible — Très élevée

Demande temporelle : À quel point le rythme de la tâche était-il pressé ou précipité ?

Très faible — Très élevée

Performance : Dans quelle mesure avez-vous réussi à accomplir ce qu'on vous avait demandé de faire ?

Parfaite — Echec

Effort : Quel effort avez-vous dû fournir pour atteindre votre niveau de performance ?

Très faible — Très élevé

Frustration : À quel point étiez-vous insécurisé, découragé, irrité, stressé et ennuyé ?

Très faible — Très élevée

Appendix E

Alternative implementation of the learning algorithm

In this appendix, we consider a simplified scenario where only the task parameters have to be learned[1]. Since the model of the robot is not considered, the experimental setup is simply composed of a physical master robot (haptic interface) and a simulated planar robot. Note that the material presented in the following is adapted from the publication (Poignonec et al. 2021) and the notations have not been fully harmonized, especially the way time dependence is denoted. As in Chapter 6, the task model is updated while it is simultaneously used to physically guide the user, allowing for *in-situ* correction of haptic guidance from user inputs only. We show that a window-based learning method is a pertinent solution to this class of problems and discuss its advantages and drawbacks.

E.1 Considered scenario and problem statement

An operator manipulates the master robot such that the follower robot follows the desired trajectory $x^d_{s,k}$. As in Chapter 5 (see Equation 5.9), the position of the robot $x_{s,k}$ is considered to be a noisy observation of the desired trajectory such that

$$x_{s,k} = x^d_{s,k} + \epsilon_{h,k}, \ \ \epsilon_{h,k} \sim \mathcal{N}(0, \Sigma_{h,k}) \tag{E.1}$$

where $\epsilon_{h,k}$ is the supposedly zero-mean error of Gaussian distribution $\mathcal{N}(0, \Sigma_{h,k})$ introduced by the operator. Let the desired trajectory be defined as a function parameterized by a set of parameters θ_g (supposedly constant) that encodes the geometry of the underlying path and a time-varying variable ψ_k that encodes the advancement along this path such that

$$x^d_{s,k} = g(\theta_g, \psi_k) \tag{E.2}$$

It should be noted that contrarily to chapters 5 and 6, the task parameters θ_g in Equation (E.2) do not encode the time parameterization, ψ_k does.

[1]The presented approach could be extended to the more general case of task and robot models correction. However, this work was actually carried out as a preliminary study of the material presented in chapters 5 and 6 such that the focus was solely on the online task registration from operator inputs.

From the robot trajectory $x_{s,k}$ resulting from the realization of the desired task by an operator, the learning problem consists in inferring the correct task parameters θ_g and a model of the time-varying variable ψ_k. This problem formulation is then similar to the one introduced in Chapter 5, except that the robot model parameters are assumed known (or that Cartesian robot pose estimation is available) and that no assumption is made about the model of ψ_k. Indeed, if we assume a constant velocity model for ψ_k such that $\psi_{k+1} = \psi_k + aT_s$ and include the execution velocity parameter a in the task parameters θ_g, we find the previous problem formulation from Chapter 5.

E.2 Proposed sliding adaptive-size window-based learning method

E.2.1 Learning the task parameters

We propose to implement the learning using a geometrical approach. If one ignores the time parameterization and rather consider ψ_k as a free parameter, the learning problem becomes similar to a non-rigid path registration problem. Recorded robot positions over a time horizon are matched with their closest point on the current estimated desired path. Under the assumption that the current task parameters estimation is correct, the computed closest points are then an estimation of the desired robot positions. Finally, we can update the estimated parameters $\hat{\theta}_{g,k}$ by minimizing the errors between robot positions and estimated desired robot positions.

To do so, the operator inputs $x_{s,k}$ are sampled periodically to form a trace of N observations spanning a finite temporal horizon. To differentiate discrete time from the indexing of the sliding window, we introduce an indexing operator $[\cdot]$. With this notation, $x_s[k] = x_{s,k}$ is the latest entry acquired at time t_k and older entries are denoted

$$x_s[k-i] = x_s(t - iT_s) \tag{E.3}$$

where $i \in [0; N-1]$ is the index within the window and T_s is the sampling period. This differentiation is necessary because some estimates related to older observations are computed at current time t_k. This is notably the case for $\hat{\psi}[k-i]$ in Equation (E.6).

The learning of the task parameters θ_g is formulated as a quadratic optimization problem, minimizing the cost function $\mathcal{L}(\theta_g)$ defined at discrete time t_k as:

$$\mathcal{L}(\theta_g) = \frac{1}{N} \sum_{i=0}^{N-1} l_i^2(\theta_g) \tag{E.4}$$

where

$$l_i(\theta_g) = x_s[k-i] - g(\theta_g, \hat{\psi}[k-i]) \tag{E.5}$$

and $\hat{\psi}[k-i]$ results from a line search performed on the path underlying the task model parameterized with the current parameters estimate $\hat{\theta}_{k-1}$ (i.e., parameters estimates computed at time t_{k-1}) such that

$$\hat{\psi}[k-i] = \arg\min_{\psi} x_s[k-i] - g(\hat{\theta}_{g,k-1}, \psi) \tag{E.6}$$

Therefore, $g(\hat{\theta}_{g,k-1}, \hat{\psi}[k-i])$ is the closest point from $x_s[k-i]$ that belongs to the path $g(\hat{\theta}_{g,k-1}, \psi)$ and $l_i(\theta_g)$, if evaluated with $\theta_g = \hat{\theta}_{g,k-1}$, is the distance between the computed closest point and $x_s[k-i]$ as illustrated in Figure E.1. Note that the $\hat{\psi}[k-i]$ values must be re-evaluated each time the task parameter estimates change, since updating the task parameters effectively modify the path on which the line search is performed, thus invalidating previous line search solutions.

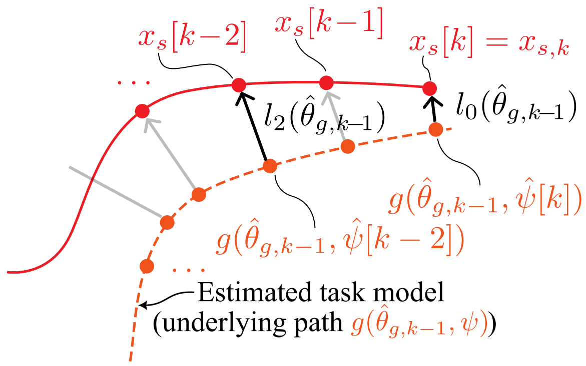


Figure E.1: Illustration of the result of the line search process for the first (latest) elements of the sliding window. The second sample is only partially annotated so as to not clutter the figure.

To minimize the cost function $\mathcal{L}(\theta_g)$ over time and consequently reduce task modeling errors, a damped Gauss-Newton optimization (Björck 1996) is implemented. This method is highly tractable for online schemes and allows the learning rate to be tuned, a desirable property for online registration-like algorithms (Hersch et al. 2012). With λ_θ the learning rate, the parameters update rule is

$$\hat{\theta}_{g,k} = \hat{\theta}_{g,k-1} - \lambda_\theta \left(J_{\theta,k}^T J_{\theta,k}\right)^{-1} J_{\theta,k}^T \begin{bmatrix} l_0(\hat{\theta}_{g,k-1}) \\ l_1(\hat{\theta}_{g,k-1}) \\ \vdots \\ l_{N-1}(\hat{\theta}_{g,k-1}) \end{bmatrix} \tag{E.7}$$

where $J_{\theta,k} \in \mathbb{R}^{N\dim(x_s)\times\dim(\theta_g)}$ is the Jacobian w.r.t. θ_g of the vector containing the residuals $l_i(\hat{\theta}_{g,k-1})$. The Jacobian matrix $J_{\theta,k}$ is evaluated using the latest task parameters estimate $\hat{\theta}_{g,k-1}$ such that

$$J_{\theta,k} = \begin{bmatrix} \dfrac{\partial l_0(\theta_g)}{\partial \theta_g} \\ \dfrac{\partial l_1(\theta_g)}{\partial \theta_g} \\ \vdots \\ \dfrac{\partial l_{N-1}(\theta_g)}{\partial \theta_g} \end{bmatrix}_{\theta_g = \hat{\theta}_{k-1}} \tag{E.8}$$

E.2.2 Online window size management

Using a window of observations raises a non-trivial question: that of its size. At startup, new samples are added to the window one at a time such that we need a way to decide whether enough data has been collected to start the learning or not. Then, the size of the window has to be managed continuously: if the window is too small, the information contained in the samples will be insufficient to learn the parameters; if the window is too large, the learning will not be able to cope with any change in the parameters. In order to adapt the size of the window online, it is necessary to have a way of evaluating if the window contains enough information.

Let γ_k be a metric computed in such a way that if $\gamma_k \leqslant \gamma_{max}$, then the current window of size N_k is sufficient for a meaningful and stable learning. The size of the window should therefore verify the inequality $\gamma_k \leqslant \gamma_{max}$, with γ_{max} a manually tuned threshold. As new observations are sampled one at a time, the ideal size for the window varies slowly. Therefore, we propose to expand or shrink the window used for learning by unitary increments:

$$\begin{cases} N_{min} \mathrel{+}= 1 & if\, \gamma_k \geqslant s\gamma_{max} \text{ and } N_{min} < N_{max} \\ N_{min} \mathrel{-}= 1 & if\, \gamma_k < s\gamma_{max} \text{ and } N_{min} > 1 \end{cases} \tag{E.9}$$

where N_{min} is the minimal size of the window verifying the criterion $\gamma_k \leqslant \gamma_{max}$ and $s < 1$ is a coefficient used to ensure a margin in the minimal window size N_{min}. Finally, the actual size N_k used for the learning is chosen such that $N_k = N_{min,k}$ and the parameters are only updated if $\gamma_k \leqslant \gamma_{max}$. The chosen metric γ_k is based on the expected covariance of the estimated desired position further along the path given the collected data. For more details about the criterion γ_k, we refer the reader to the publication (Poignonec et al. 2021).

E.2.3 Adaptive haptic guidance

Note to the reader: As the haptic guidance is generated at a much higher rate than the learning (i.e., 1 kHz vs. 10 Hz), continuous time will be used in the following equations so as to not introduce a second discrete time. However, as the haptic guidance is implemented on numerical hardware, the variables introduced in the following remain discrete in nature.

The proposed online parameters learning method can be used to implement an adaptive haptic guidance similarly to what was done in Chapter 5. To do so, the reference for master side haptic guidance is computed as

$$\begin{aligned} x_m^g(t) &= \mathcal{M}^{-1}\left(\hat{x}_s^d(t)\right) \\ &= \mathcal{M}^{-1}\left(g(\hat{\theta}_g(t), \hat{\psi}(t))\right) \end{aligned} \tag{E.10}$$

However, the time parameterization of the trajectory $\hat{x}_s^d(t)$ is not learned by the proposed learning method such that the model $\hat{\psi}(t)$ has to be estimated in some way. The most straightforward option consists in computing at all time the closest point on the estimated desired path such that

$$\hat{\psi}(t) = \psi_c(t) \tag{E.11}$$

where

$$\psi_c(t) = \arg\min_{\psi} x_s(t) - g(\hat{\theta}_g(t), \psi) \tag{E.12}$$

The haptic guidance would then be a classical virtual fixture (Bowyer et al. 2014), albeit an adaptative one. It should be noted that using Equation (E.12), the movements of the operator are completely free along the direction tangent to the path. Although this is desirable in some cases, this might also lead to a discontinuous guidance if $\hat{\psi}(t)$ varies suddenly, which is a common phenomenon with this type of line search optimization to find the closest point on a path or surface (Pegna et al. 1996; Ko et al. 2014).

Therefore, we propose instead to smooth the evolution of $\hat{\psi}(t)$, which also has the added benefit of smoothing the operator's movements. To that end, we introduce a parametric model of the time parameterization $\hat{\psi}(t) = a(t)t + b(t)$ and the error $e_\psi(t) = \psi_c(t) - \hat{\psi}(t)$, where $\psi_c(t)$ is computed using Equation (E.12). The update rule for the time parameters $a(t)$ and $b(t)$ uses a damped Gauss-Newton method with learning factor λ_ψ as follows:

$$\hat{\psi}(t) = a(t)t + b(t) \tag{E.13}$$

$$\begin{bmatrix} \dot{a}(t) \\ \dot{b}(t) \end{bmatrix} = \lambda_\psi \left(J_\psi(t)^T J_\psi(t) \right)^{-1} J_\psi(t)^T \begin{bmatrix} e_\psi(t) \\ \dot{e}_\psi(t) \end{bmatrix} \tag{E.14}$$

where

$$J_\psi(t) = \begin{bmatrix} \frac{\partial \psi}{\partial a} & \frac{\partial \psi}{\partial b} \\ \frac{\partial \dot{\psi}}{\partial a} & \frac{\partial \dot{\psi}}{\partial b} \end{bmatrix} \tag{E.15}$$

E.3 Experimentation with an operator in-the-loop

The sliding window learning method is experimentally validated with an operator in-the-loop on a 2D path-following task. The master robot is a haptic interface and a simulated robot and environment serves as the teleoperated follower robot (see Figure E.2). The aim is to show that the proposed sliding window learning method is capable of correcting an initially inaccurate task registration online and from operator actions alone.

E.3.1 Experimental setup

The master robot is a 3DOF haptic interface Omega 3 from Force Dimension. The simulated robot has no dynamic and the position-position teleoperation mapping $\mathcal{M}(\cdot)$ is a one-to-one mapping such that we denote $x(t) = x_m(t) = x_s(t)$ the position of both the master and follower robot (that are here exactly the same). A master side haptic

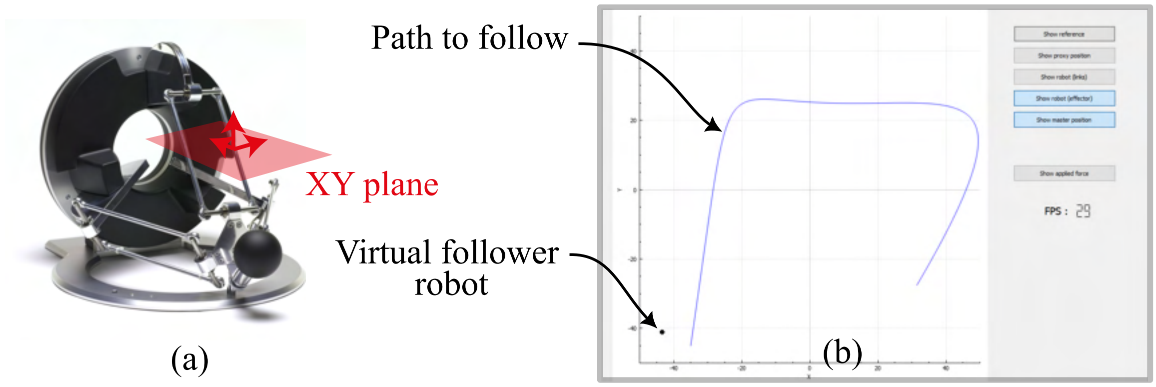


Figure E.2: Experimental setup: (a) 3 DOF haptic interface manipulated by the operator. (b) Visual feedback during the experiment.

guidance is generated from the guidance reference $x_m^g(t)$ defined by equations (E.10) and (E.13). In order to smooth the transitions introduced by the update of $\hat{\theta}_k$, the estimated parameters used for haptic guidance $\hat{\theta}(t)$ are filtered (see Figure E.3) by a first order filter with a time constant of 1 s. Since the task is two-dimensional, the operators movements are constrained to the XY plane through a higher guidance stiffness $k_{plane} = 1500$ N in the Z direction (see Figure E.2). The haptic guidance stiffness K_d is set to $\text{diag}(K_d) = [k_g, k_g, k_{plane}]$, where $k_g = 200$ N. The haptic loop runs at 1 kHz and the learning loop runs at 10 Hz ($T_s = 0.1$ s). The learning process is implemented as described in Figure E.3, with hyperparameters $\lambda_\theta = 0.001$, and $\lambda_\theta = 0.2$, $N_{max} = 200$. The other hyperparameters used for the window size management can be found in the publication (Poignonec et al. 2021).

The experiment is as follows. A path is displayed on the screen, along with a dot representing the position of the follower robot (see Figure E.2). The user is asked to manipulate the haptic interface to follow the path with the virtual "robot," with no constraint on the velocity. The path displayed on the screen is considered to define the ground truth for the geometrical part of the desired task $x_s^d(t)$ such that

$$g(\theta_g, \psi) = \begin{bmatrix} t_x \\ t_y \\ 0 \end{bmatrix} + \begin{bmatrix} \cos r_z & -\sin r_z & 0 \\ \sin r_z & \cos r_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \Gamma(\psi) \tag{E.16}$$

where $\theta_g = [r_z, t_x, ty]$ is the vector of task parameters and $\Gamma(\psi)$ is a known path (b-spine) in the XY plane, parameterized such that $\|\dot{\Gamma}(\psi)\|$ is constant and $\psi \in [0; 1]$ for the considered range of motions.

E.3.2 Results

As visible in Figure E.4, new observations are added to the sliding window as the operator performs the task. The size of the sliding window increases slowly at startup and then stabilizes around $N_k = 55$ (see Figure E.7). We did not detail the computation of the criterion γ_k, but it should be noted that the window continuously expands and shrinks to keep γ_k in a predefined range (Poignonec et al. 2021). After some time (about

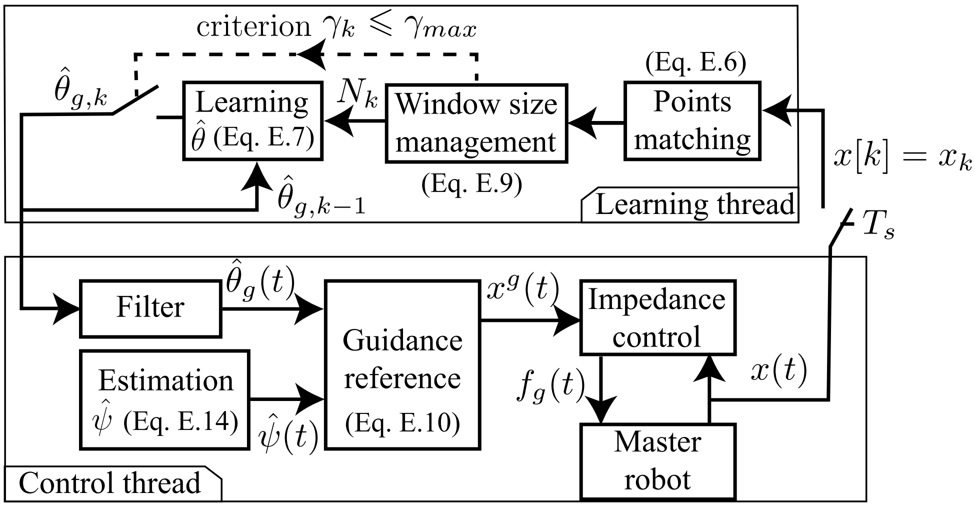


Figure E.3: Overview of the proposed method and experimental implementation.

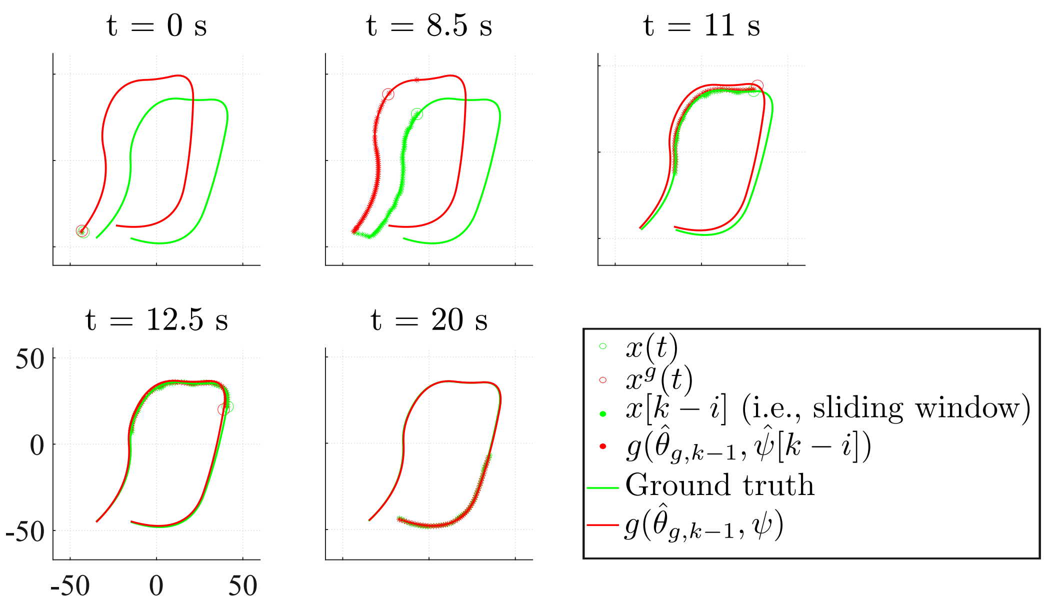


Figure E.4: Snapshots of the different positions of interest at different times throughout the experiment. All in mm, the scale displayed in the bottom-left figure.

9 s), the criterion used to evaluate the quality of the information is met and the learning starts. The estimated parameters then quickly converge towards their correct values such that the registration is corrected (see Figure E.4). The evolution of individual parameters is reported in Figure E.5, where one can observe that the estimation errors converge to very small values. At steady state, the RMSE of the parameter estimation over the sliding window computed as

$$\mathrm{RMSE}(\hat{\theta}_{g,k}) = \sqrt{\mathcal{L}_\theta(\hat{\theta}_{g,k})} \tag{E.17}$$

is reduced to 0.45 mm. This is a significant reduction as the $\mathrm{RMSE}(\hat{\theta}_{g,k})$ reaches as high as 14 mm (i.e., 98% decrease of the task prediction errors).

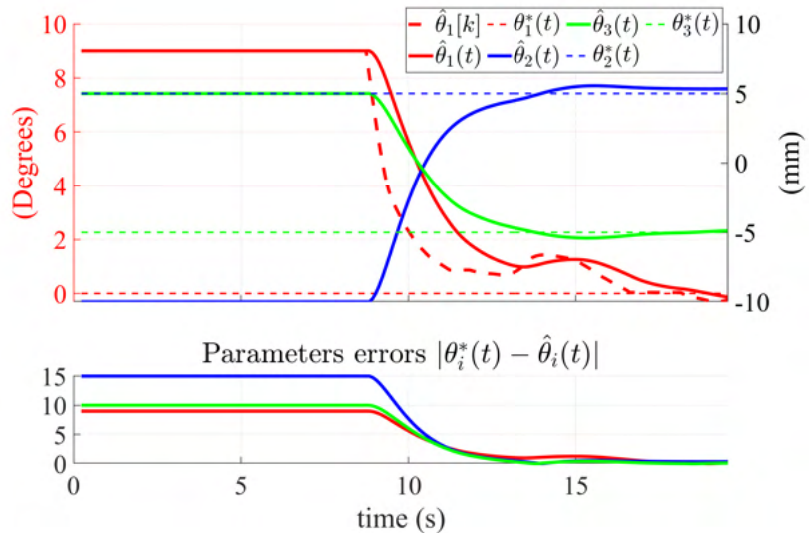


Figure E.5: Top: evolution of the estimated parameters and comparison with the ground truth. The notation $\theta_{g,i}(t)$ refers to the i$^{\text{th}}$ element of $\theta_g(t)$ such that $\theta_{g,1}(t) = r_z$, $\theta_{g,2}(t) = t_x$, and $\theta_{g,3}(t) = t_y$. Only the parameters $\hat{\theta}_g(t)$ used for generating the guidance reference are shown, except for $\hat{\theta}_{g,1,k}$, included to show the effect of the filter. The correct parameter values are displayed using dotted lines. Bottom: absolute error on estimated parameters used for guidance $|\theta_{g,i}(t) - \hat{\theta}_{g,i}(t)|$.

The tracking of the time parameter $\psi(t)$ (see Figure E.6) behaves as expected: it smooths the results obtained from the line search (see Equation E.12). For instance, at $t = 8.5$ s, the line search results in a discontinuity visible in Figure E.6, but also in Figure E.4 (upper right of the vignette $t = 8.5$ s). Such a discontinuity would be experienced as jerk by the operator if $\psi_c(t)$ was directly used to generate the guidance reference, but the time parameters tracking effectively filters out these artifacts (see Figure E.6).

E.4 Conclusions

The approach developed in the previous can successfully correct the geometrical parameters of the task (i.e., all but the time parameters) online and efficiently. These parameters can then be used to generate a haptic guidance, provided that the time parameterization is reintroduced through either the estimation of time parameters or the implementation of a virtual fixture. We chose to do the former, allowing to smooth the movements of the operator along the path. Nonetheless, it should be noted that the independent time parameters learning could be exploited to seamlessly implement other guidance strategies such as classical virtual fixture or a so-called look-ahead guidance.

This approach, although not as generic as the one proposed in Chapter 5, has the advantage of learning the parameters from a window of observations and not only from the latest observation. This is an interesting feature that can be exploited to perform online analysis (e.g., for parameters identifiability) of the data actually collected.

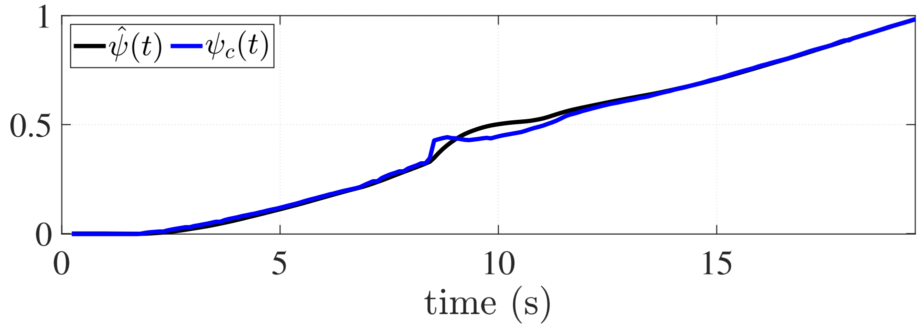


Figure E.6: Estimated values for $\hat{\psi}(t)$ and $\psi_c(t)$.

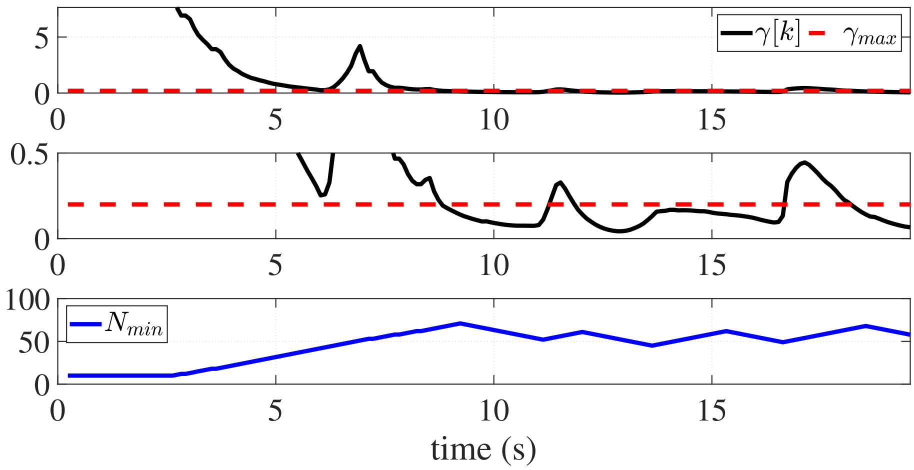


Figure E.7: Top: criterion γ_k and threshold γ_{max} during the experiment, displayed on a log-scale. Center: zoom of the figure at the top. Bottom: the window shrinks or expands to keep the criterion within a predetermined range.

Appendix F

Extended abstract (french)

F.1 Introduction

F.1.1 La chirurgie minimalement invasive

Grâce aux progrès de la médecine au cours des dernières décennies, de nouvelles approches de moins en moins invasives sont apparues dans nos hôpitaux. Notamment, la miniaturisation des technologies de capture vidéo et des outils chirurgicaux ont permis le développement de la cœlioscopie pour accéder aux cavités abdominales à travers de petites incisions à la surface. De façon similaire, les progrès en endoscopie flexible ont permis la réalisation d'opérations chirurgicales dites « sans cicatrice » qui consistent à opérer en accédant à la zone d'intérêt par des orifices naturels. Ces nouvelles approches minimalement invasives améliorent grandement la qualité des chirurgies, avec des patients qui ont un temps de récupération plus court, des risques postopératoires réduits, ainsi que des cicatrices moins visibles. En revanche, ce passage depuis la chirurgie ouverte vers des techniques moins invasives a également fortement dégradé le confort et l'ergonomie des chirurgies pour les praticiens. Notamment, leur champ de vue s'est vu réduit et ils opèrent maintenant avec des outils miniatures et souvent complexes à manipuler, qui plus est, avec des postures inconfortables durant des chirurgies plus longues.

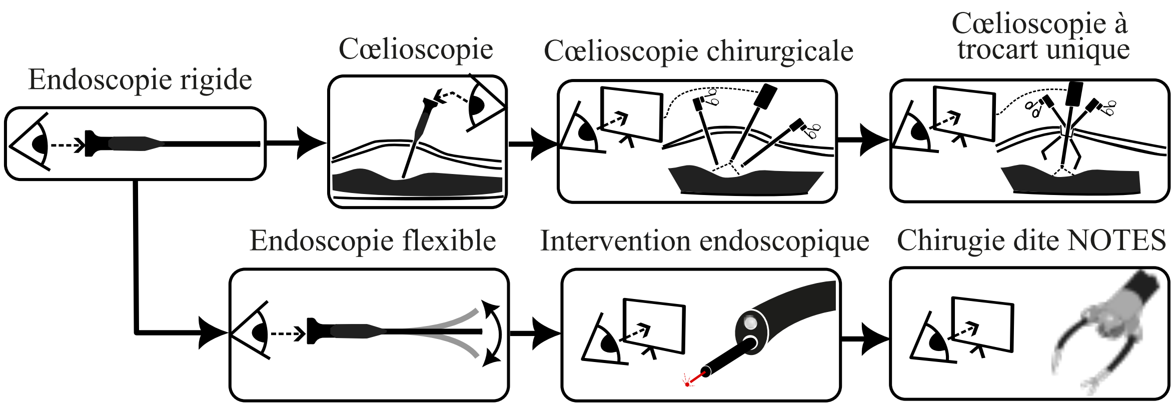


FIGURE F.1 : Illustration de l'évolution des techniques endoscopiques.

Pour répondre à ces problèmes, les technologies de chirurgie robotisée se sont développées en parallèle pour apporter une assistance à la chirurgie minimalement invasive, d'abord avec des assistants robotiques à la cœlioscopie, puis avec des solutions variées permettant de télémanipuler des outils chirurgicaux complexes à partir d'une seule console maitre. Maintenant bien répandue dans les hôpitaux du monde entier, cette assistance robotique permet d'améliorer la dextérité, la précision et le confort du chirurgien. Les robots pour l'assistance à la chirurgie minimalement invasive sont conçus pour répliquer exactement les gestes effectués par le chirurgien sur la console maitre, tel que tout mouvement est initié par l'humain et, éventuellement, mis à l'échelle. Cette télémanipulation transparente des robots permet aux chirurgiens de maitriser tous les aspects de l'acte chirurgical, mais dans certains cas une aide plus active de la part du robot serait bénéfique.

F.1.2 Vers une assistance active à la chirurgie robotisée

Une plus grande autonomie des dispositifs robotiques permet de réduire la fatigue mentale et physique de l'opérateur ainsi que d'augmenter la procédure par des technologies d'imagerie et d'IA, ou par l'exécution de gestes qui seraient complexes, voire impossibles pour un humain. En revanche, dans le contexte de la robotique chirurgicale, les contraintes légales ne permettent pas à ce jour de passer le cap vers une autonomie totale et la technologie n'est de toute façon pas suffisamment mature. Des modes semi-automatiques restent toutefois pertinents, par exemple dans le cas de tâches répétitives nécessitant peu voire pas de prise de décision, comme la suture ou la palpation. Ces sous-tâches pourraient être automatisées, partiellement ou totalement, afin d'améliorer la répétabilité du geste tout en réduisant la charge cognitive pour le chirurgien. Cependant, ce type d'automatisation n'est à ce jour utilisée que dans des domaines où la tâche chirurgicale peut être définie et planifiée précisément, par exemple en neurochirurgie ou en chirurgie orthopédique où les tâches peuvent être définies à partir d'images préopératoires, puis recalées à l'aide de marqueurs. Ce n'est pas le cas avec les chirurgies de l'appareil digestif ou des cavités abdominales, notamment,car le recalage de la tâche doit se faire à partir d'images endoscopiques d'un environnement non structuré et de surcroit prône aux déformations.

La commande partagée pour une collaboration robot-chirurgien

Dans ce contexte, une assistance au geste qui laisse l'autorité finale au chirurgien est intéressante, vu qu'elle permet d'assister tout en laissant le chirurgien prendre les décisions et est alors plus robuste aux erreurs de modélisation comme le recalage. La commande partagée consiste à commander le robot à travers une collaboration avec un système d'assistance qui a une connaissance, parfois partielle, de la tâche à réaliser. Une méthode particulièrement pertinente dans le cas d'une opération chirurgicale est la commande partagée dite « haptique », qui consiste à communiquer au chirurgien les intentions du système d'assistance à travers des efforts appliqués sur la console maitre (voir Figure F.2). Cette approche permet de restreindre les zones accessibles par le robot ou de guider le geste de façon continue à travers l'application de forces de guidage. Cependant le système d'assistance, bien qu'il n'ait pas l'autorité finale, doit tout de

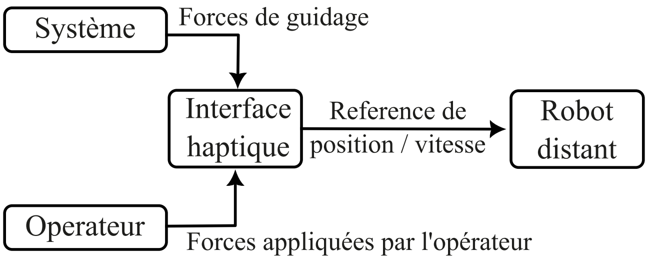
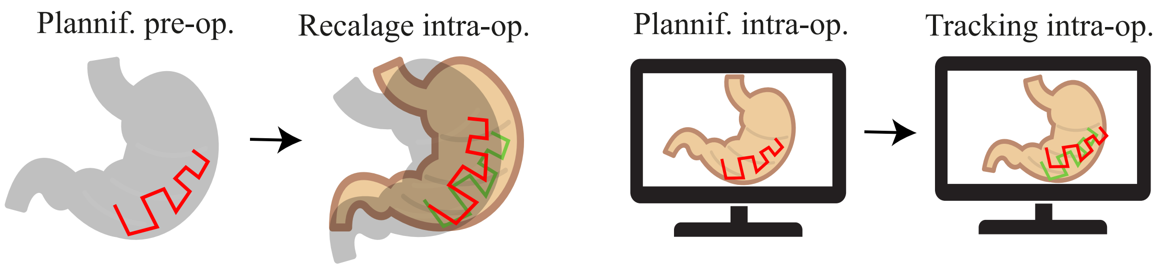


FIGURE F.2 : Illustration (simplifiée) de la commande partagée haptique.

même avoir une représentation correcte de son environnement pour pouvoir apporter une aide effective.

Difficultés rencontrées par un système d'assistance

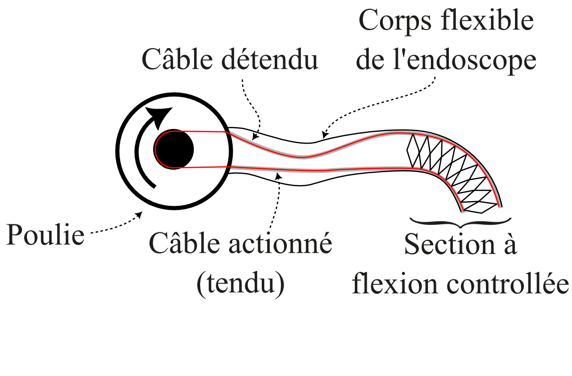
L'environnement chirurgical est par nature déformable, hautement complexe et dépendant du patient, ce qui rend la planification en ligne de la tâche complexe et peu robuste. La planification peut se faire à partir de données préopératoires, mais dans ce cas il reste à recaler la tâche dans le repère de référence du robot, avec les mêmes difficultés. De plus, hormis pour des scénarios où les tissus considérés sont rigides, une trajectoire robotique définie à partir de données préopératoires devra être corrigée lors de la procédure pour tenir compte des déformations. Même dans le cas où une planification peut être effectuée à partir d'images ou de données intra-opératoires (caméra embarquée, OCT, etc.), les conditions cliniques telles que l'éclairage non homogène de la scène, les réflexions ou les occlusions rendent la trajectoire prône aux erreurs. Ces deux cas de figure sont illustrés par la Figure F.4.



(a) (b)

FIGURE F.3 : Illustration des erreurs de recalage *in-situ* d'une tâche robotique causées par une erreur de recalage entre preopératif et intra-opératif (a) ou bien introduites peu à peu par des erreurs dans le tracking de l'environnement (b).

Aux challenges de modélisation de la tâche robotique viennent aussi s'ajouter ceux relevant de son exécution. Les robots chirurgicaux souffrent souvent de problèmes de positionnement absolu qui, bien que n'impactant pas les performances quand un chirurgien téléopère, posent problème pour des modes automatiques ou des stratégies d'assistance. Cela est causé par le manque de capteurs distaux, mais aussi par des erreurs de recalages entre la caméra et les outils robotisés. Un des cas les plus pathologiques est celui du jeu mécanique parfois présent dans la transmission, typiquement dans le cas des transmissions à câbles utilisées en endoscopie flexible (voir Figure F.4a). Non seulement ces jeux peuvent introduire des erreurs de positionnement conséquentes (voir Figure F.4b), mais ils sont aussi ressentis par le chirurgien comme des latences dans la commande du robot. Le jeu mécanique peut être compensé par la commande, mais un modèle de la non-linéarité doit être estimé au préalable.



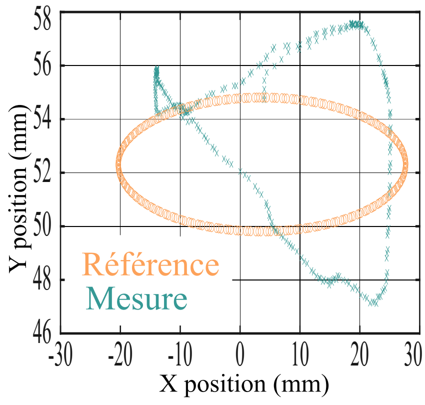


FIGURE F.4 : (a) Illustration de la transmission à câbles utilisée par beaucoup d'endoscopes flexibles, la figure est adaptée de (BARDOU et al. 2012). (b) Si le jeu mécanique n'est pas compensé, des erreurs de positionnement de l'effecteur du robot apparaissent (ALELUIA PORTO 2021).

Les erreurs de modélisation de la tâche et de la géométrie du robot impactent les modes automatiques de façon évidente, mais les effets sur des stratégies d'assistance sont similaires. Par exemple, dans le cas où un guidage haptique est implémenté pour assister le geste médical, le chirurgien peut compenser les erreurs de modélisation. Néanmoins, il ou elle sera guidée vers une trajectoire incorrecte, ce qui sera une source de conflit entre le chirurgien et le système. Idéalement, le système d'assistance devrait pouvoir être corrigé *in-situ*, voire en ligne lors de son utilisation afin d'améliorer en continu la qualité de l'assistance.

F.1.3 Objectifs de la thèse

Nos travaux visent donc à développer des moyens d'affiner en cours de chirurgie (in-situ) les modèles internes du système d'assistance, ce qui comprend entre autres le recalage de la tâche chirurgicale. Un modèle correct de la géométrie du robot est aussi nécessaire, à la fois pour permettre un positionnement précis et pour générer une assistance effective. Les objectifs sont alors multiples :

1. apprendre in-situ les modèles de recalage de tâche et de la géométrie/dynamique du robot ;

2. utiliser ces modèles pour apporter une assistance au geste chirurgical ;

3. continuer d'améliorer les modèles durant leur utilisation afin d'améliorer le guidage en continu.

Plus généralement, cette thèse s'intéresse à la mise à jour des modèles précédents afin d'améliorer l'assistance robotisée aux gestes chirurgicaux, avec un intérêt particulier pour les applications endoscopiques.

F.2 Organisation du document et contributions

De façon générale, les travaux présentés dans ce document de thèse traitent de l'apprentissage des modèles nécessaires à l'assistance de gestes médicaux chirurgicaux. Le travail se divise en deux axes principaux qui adressent deux défis importants de la robotique pour la chirurgie minimalement invasive : la compensation du jeu mécanique présent dans les endoscopes flexibles ou autres robots actionnés par câbles et l'apprentissage de modèles pour la commande partagée des robots chirurgicaux. De fait, le document lui-même est organisé en deux parties qui sont décrites ci-dessous.

F.2.1 Identification du jeu mécanique des endoscopes flexibles robotisés

Nous proposons de nouvelles méthodes pour l'identification *in-situ* du modèle du jeu mécanique dans la transmission à câbles des robots endoscopiques flexibles, un problème critique aussi bien lors de la téléopération directe ou de l'exécution de trajectoires automatiques. Le jeu mécanique est principalement causé par le jeu dans les câbles antagonistes, leur élasticité et la déformation de l'endoscope lui-même (voir Figure F.4a). Le jeu mécanique complexifie la manipulation des systèmes endoscopiques, car il est ressenti par le chirurgien comme un retard entre les déplacements proximaux et distaux. De plus, la présence d'un jeu mécanique dégrade la précision du positionnement et rend l'implémentation de modes de commande automatiques complexes, voire impossibles. De fait, il convient d'estimer ces jeux afin de les compenser. En revanche, le comportement peut évoluer au cours du temps et avec la manipulation de l'endoscope (déformation du corps de l'endoscope) tel qu'une identification in-situ du modèle du jeu mécanique est préférable.

Dans le chapitre 2, nous présentons la littérature scientifique autour de l'identification des modèles d'endoscopes. Puis, nous proposons une approche permettant une identification en ligne des modèles de jeu mécanique adaptée au cas de l'endoscopie flexible. L'approche est basée sur le principe de filtrage de Kalman discontinu, ce qui permet une formulation générale du problème. Nous modifions la méthode originale de Chatzis et al. (2017b) afin de simplifier le réglage du filtre et d'améliorer ses performances, notamment à travers l'ajout d'un mécanisme d'oubli exponentiel. La méthode est évaluée sur une simulation d'un outil endoscopique à trois degrés de liberté.

L'approche proposée dans le chapitre 2, ainsi que la plupart des méthodes d'identification existantes, nécessite une mesure directe de la position distale du robot. Cependant, il existe des cas où une telle mesure est impossible, typiquement pour des configurations dites eye-in-hand, c'est-à-dire quand la caméra est embarquée sur l'organe terminal du robot (voir Figure F.5). Nous proposons donc dans le chapitre 3 une méthode d'identification du jeu mécanique inspirée des notions de localisation et cartographie simultanées (SLAM, pour "simultaneous localization and mapping") qui permet d'effectuer l'identification *in-situ* du jeu mécanique à partir des seules images endoscopiques. La méthode proposée repose sur une modélisation particulière de jeu mécanique qui permet de séparer la largeur du jeu de la relation mathématique globale. Afin de résoudre le problème, l'approche proposée dans le chapitre 2 est utilisée. Nous validons cette méthode expérimentalement sur un endoscope clinique dont la configuration eye-in-hand ne permet pas l'utilisation de méthodes plus classiques.

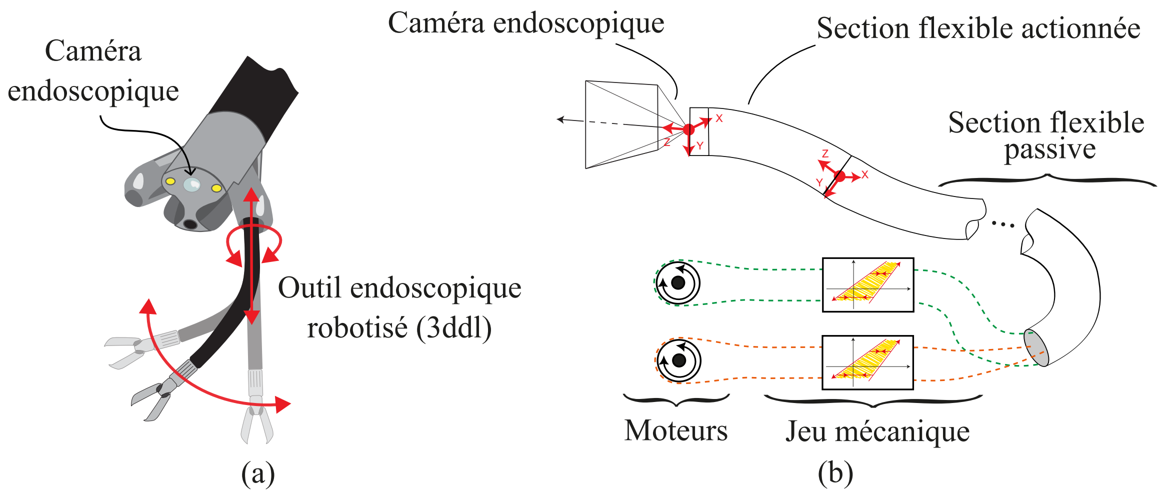


FIGURE F.5 : (a) Illustration de l'outil endoscopique à trois degrés de liberté (ddl) considéré dans les chapitres 2 et 4. (b) Illustration d'un endoscope flexible robotisé dans une configuration dite eye-in-hand tel que considéré dans le chapitre 3.

Enfin, dans le chapitre 4, nous explorons comment la modélisation alternative du jeu mécanique consistant à rendre la largeur du jeu indépendante du reste peut être exploitée pour apprendre (partiellement) le modèle à partir de la simple détection binaire du mouvement dans l'image endoscopique. Une fois la largeur de jeu identifiée, le jeu mécanique peut être compensé par la commande. En revanche, ce modèle partiel ne permet pas un positionnement correct du robot. Nous proposons alors une identification de la non-linéarité restante basée sur des méthodes plus classiques d'estimation de pose du robot. L'originalité consiste à combiner cette information avec le modèle de largeur de jeu estimé par détection de mouvement afin que les deux phases d'identification puissent être utilisées de façon indépendante. Cette approche fut validée sur une plateforme d'endoscopie robotisée développée au sein du laboratoire ICube.

Pour résumé, nos contributions à l'identification *in-situ* de ces jeux mécaniques sont les suivantes :

- nous proposons un cadre unifié et général pour l'estimation du jeu mécanique adapté à l'identification en ligne de modèles d'endoscopie flexible. La méthode et les résultats seront publiés dans (POIGNONEC et al. 2023a) ;

- nous proposons une nouvelle approche pour l'estimation du jeu mécanique basée image et adaptée aux endoscopes « eye-in-hand ». Ce travail fait l'objet d'une publication en préparation ;

- enfin, nous développons une nouvelle approche pour l'estimation *in-situ* du jeu mécanique qui ne nécessite pas une estimation intermédiaire de la pose du robot contrairement aux méthodes existantes. L'approche est basée sur la détection du mouvement dans l'image endoscopique. Ces travaux furent publiés dans (Poignonec et al. 2020).

F.2.2 Apprentissage de modèles pour la commande partagée

Dans le cadre de la commande haptique partagée, nous proposons des algorithmes pour corriger le recalage d'une tâche et autres erreurs de modélisation à partir d'informations extraites de la présence du chirurgien qui reste présent dans la boucle. Cette approche ne repose sur aucun capteur extéroceptif (comme des caméras) dont les mesures pourraient devenir indisponibles ou erronées, mais utilise plutôt l'opérateur comme source d'information. L'apprentissage en ligne des modèles est utilisé pour implémenter un guidage haptique adaptatif qui s'améliore à l'utilisation : si le guidage est initialement erroné, l'opérateur va résister en appliquant des forces sur la console maitre (interface haptique). Cela cause la mise à jour des modèles puis, mécaniquement, la réduction des erreurs de guidage.

Dans le chapitre 5, après avoir introduit la littérature scientifique relevant de l'apprentissage de modèles pour le guidage et la coopération humain-robot, nous proposons une approche d'apprentissage en ligne pour corriger simultanément les paramètres des modèles du robot et de la tâche. Une implémentation récursive basée sur le filtrage de Kalman est proposée et validée à travers des simulations. Afin de pouvoir appliquer par la suite la méthode à des scénarios avec un humain dans la boucle, certains aspects pratiques sont traités. Entre autres, nous proposons une méthode pour améliorer le tracking des paramètres temporels pour que le filtre puisse s'adapter à des vitesses d'exécution variables imposées par l'opérateur. Les chapitres 6 et 7 sont alors dédiés à la validation expérimentale.

Dans le chapitre 6, la méthode proposée dans le chapitre 5 est validée expérimentalement avec une tâche de téléopération générique (dessin) sur une plateforme robotique composée d'un robot industriel et d'une interface haptique. Dans le scénario envisagé, une tâche a été définie à partir de données preopératoire, mais son recalage dans l'espace opérationel du robot est incorrecte. De plus, il y a une incertitude sur la position de l'outil par rapport à l'effecteur du robot (voir Figure F.6). Notre méthode d'apprentissage est utilisée pour apprendre en ligne le recalage de la tâche et de l'outil à partir des actions de l'utilisateur seules. La performance de l'apprentissage et du guidage haptique sont évaluées dans différents scénarios afin de démontrer l'applicabilité de l'approche.

Enfin, dans le chapitre 7, les résultats d'une étude avec participants ($N = 12$) sont présentés. L'étude vise à valider expérimentalement la méthode d'apprentissage et le guidage, à la fois quantitativement à travers des métriques comme la précision ou la régularité du geste, mais aussi qualitativement. À cette fin, des questions permettant d'évaluer des notions subjectives telles que la pénibilité ou la sensation de contrôle sont

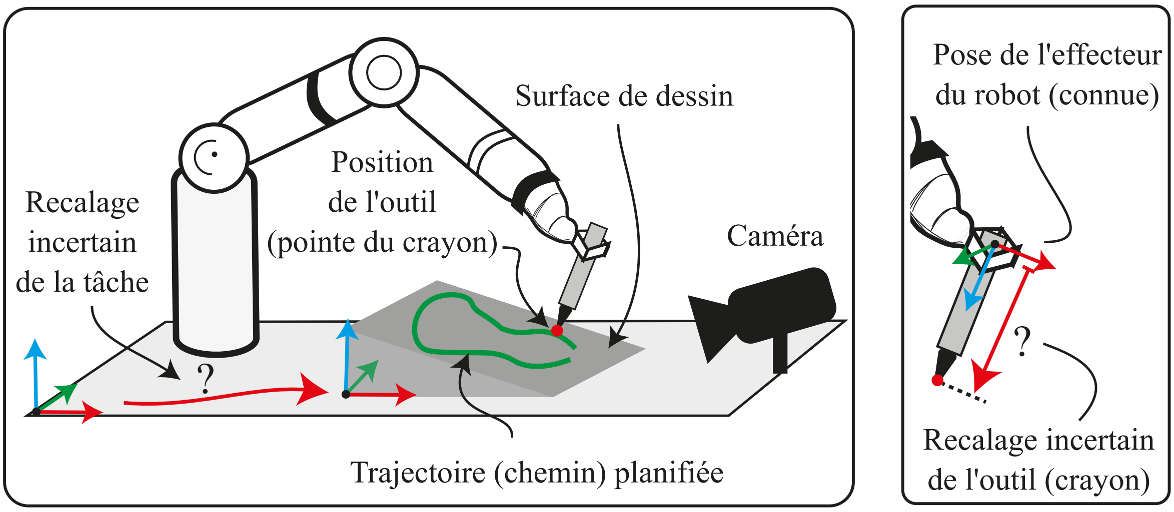


FIGURE F.6 : Illustration du scénario considéré pour les validations expérimentales des chapitres 6 et 7.

posées aux participants à travers des questionnaires. Globalement, l'étude confirme qu'un guidage adaptatif permet d'améliorer les performances par rapport à un guidage qui ne le serait pas. De plus, nous montrons qu'un guidage incorrect est comparable à une absence de guidage en termes de performances, tous deux bien inférieur au cas avec guidage adaptatif.

Pour résumé, nos contributions à l'apprentissage en ligne de modèles pour le guidage haptique les suivantes :

- nous proposons une nouvelle méthode la correction simultanée du recalage de la tâche et du modèle géométrique de robot. Il s'agit d'un cadre théorique unifié qui permet la correction en ligne des paramètres des modèles à partir de l'observation des actions de l'opérateur. Ce travail fait l'objet d'une publication en préparation ;
- nous présentons des résultats complets concernant l'effet du guidage haptique adaptatif sur la performance de l'opérateur et la pénibilité perçue recueillis lors d'une étude avec participants. Les principaux résultats font l'objet d'une publication en préparation ;
- nous présentons également une méthode d'apprentissage basée sur une optimisation sur une fenêtre glissant de mesures pour réaliser la correction du recalage de la tâche à partir des actions de l'utilisateur. Ce travail a fait l'objet d'une publication (POIGNONEC et al. 2021).

F.3 Conclusions

Dans la première partie, nous proposons plusieurs méthodes complémentaires pour l'identification *in situ* du modèle de jeu mécanique présent dans les transmissions à câbles des endoscopes flexibles. Par rapport aux méthodes existantes, les approches proposées permettent l'utilisation d'un modèle de jeu plus complexe, ce qui est pertinent pour les scénarios cliniques. Toutes les méthodes proposées sont très génériques et peuvent être appliquées à différentes architectures de robots ainsi qu'à différents capteurs intraopératoires.

Dans la seconde partie, nous proposons de nouvelles méthodes de guidage adaptatifs adaptées à la télérobotique. La principale contribution est le développement d'une méthode d'apprentissage en ligne des paramètres de la tâche et du robot qui utilise l'opérateur dans la boucle comme source d'information. La méthode est applicable lorsque les modèles sont connus, mais que les valeurs de certains paramètres sous-jacents ne le sont pas. C'est typiquement le cas lorsqu'une tâche a été définie, mais que les paramètres doivent être adaptés à l'environnement réel durant l'exécution. Différents défis ont été identifiés et des solutions ont été proposées. Les résultats expérimentaux soutiennent l'idée qu'un guidage adaptatif est préférable à son équivalent non adaptatif lorsque les modèles sont initialement imprécis. Ces méthodes constituent alors un bon compromis pour exploiter une planification préopératoire tout en laissant le contrôle final au chirurgien.

Bibliography

Aarno, D., S. Ekvall, and D. Kragic (2005). "Adaptive Virtual Fixtures for Machine-Assisted Teleoperation Tasks". *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 897–903 (cit. on p. 75).

Abbink, D. A., T. Carlson, M. Mulder, J. C. F. de Winter, F. Aminravan, T. L. Gibo, and E. R. Boer (2018). "A Topology of Shared Control Systems—Finding Common Ground in Diversity". *IEEE Transactions on Human-Machine Systems* 48.5, pp. 509–525 (cit. on p. 10).

Abbink, D. A., M. Mulder, and E. R. Boer (2012). "Haptic shared control: smoothly shifting control authority?" *Cognition, Technology & Work* 14.1, pp. 19–28 (cit. on pp. 9–11, 74).

Abbott, D. J., C. Becke, R. I. Rothstein, and W. J. Peine (2007). "Design of an endoluminal NOTES robotic system". *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 410–416 (cit. on pp. 4, 5).

Abi-Farraj, F., C. Pacchierotti, and P. R. Giordano (2018). "User Evaluation of a Haptic-Enabled Shared-Control Approach for Robotic Telemanipulation". *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–9 (cit. on p. 122).

Accuray (2022). *Commercial website of Accuray*. URL: `https://www.accuray.com/` (visited on 12/15/2022) (cit. on p. 6).

Aleluia Porto, R. (2021). "Modelling and control of robotic flexible endoscopes using machine learning". These de doctorat. Strasbourg (cit. on pp. 8, 37, 156).

Aleluia Porto, R., F. Nageotte, P. Zanne, and M. de Mathelin (2019). "Position control of cable-driven flexible instruments by combining machine learning and kinematic analysis". *IEEE International Conference on Robotics and Automation (ICRA)* (cit. on pp. 9, 21, 22, 57, 59–61).

Attanasio, A., B. Scaglioni, E. De Momi, P. Fiorini, and P. Valdastri (2021). "Autonomy in Surgical Robotics". *Annual Review of Control, Robotics, and Autonomous Systems* 4.1, pp. 651–679 (cit. on p. 6).

Baek, D., J.-H. Seo, J. Kim, and D.-S. Kwon (2020). "Hysteresis Compensator With Learning-Based Hybrid Joint Angle Estimation for Flexible Surgery Robots". *IEEE Robotics and Automation Letters* 5.4, pp. 6837–6844 (cit. on pp. 23, 35, 61, 66).

Bajcsy, A., D. P. Losey, M. K. O'Malley, and A. D. Dragan (2017). "Learning robot objectives from physical human interaction". *Proceedings of Machine Learning Research* 78, pp. 217–226 (cit. on p. 76).

Baksic, P., H. Courtecuisse, and B. Bayle (2021). "Shared control strategy for needle insertion into deformable tissue using inverse Finite Element simulation". *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12442–12448 (cit. on p. 10).

Balasubramanian, S., A. Melendez-Calderon, A. Roby-Brami, and E. Burdet (2015). "On the analysis of movement smoothness". *Journal of NeuroEngineering and Rehabilitation* 12 (cit. on pp. 115, 118).

Bardaro, S. J. and L. Swanström (2006). "Development of advanced endoscopes for Natural Orifice Transluminal Endoscopic Surgery (NOTES)". *Minimally Invasive Therapy & Allied Technologies* 15.6, pp. 378–383 (cit. on p. 3).

Bardou, B., F. Nageotte, P. Zanne, and M. d. Mathelin (2012). "Improvements in the control of a flexible endoscopic system". *2012 IEEE International Conference on Robotics and Automation*, pp. 3725–3732 (cit. on pp. 8, 9, 22, 61, 156).

Bednarczyk, M., H. Omran, and B. Bayle (2020). "Passivity Filter for Variable Impedance Control". *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7159–7164 (cit. on p. 121).

Björck, A. (1996). *Numerical methods for least squares problems*. SIAM (cit. on p. 146).

Bobu, A., A. Bajcsy, J. F. Fisac, S. Deglurkar, and A. D. Dragan (2020). "Quantifying hypothesis space misspecification in learning from human–robot demonstrations and physical corrections". *IEEE Transactions on Robotics* 36.3, pp. 835–854 (cit. on p. 76).

Bowyer, S. A., B. L. Davies, and F. Rodriguez y Baena (2014). "Active Constraints/Virtual Fixtures: A Survey". *IEEE Transactions on Robotics* 30.1, pp. 138–157 (cit. on pp. 10, 105, 148).

Broad, A., I. Abraham, T. Murphey, and B. Argall (2020). "Data-driven Koopman operators for model-based shared control of human–machine systems". *The International Journal of Robotics Research* 39.9, pp. 1178–1195 (cit. on p. 76).

Cabras, P., F. Nageotte, P. Zanne, and C. Doignon (2017). "An adaptive and fully automatic method for estimating the 3D position of bendable instruments using endoscopic images". *The International Journal of Medical Robotics and Computer Assisted Surgery* 13.4 (cit. on pp. 35, 59, 61, 65, 66).

Chatzis, M. N. and E. N. Chatzi (2017a). "A Discontinuous Unscented Kalman Filter for Non-Smooth Dynamic Problems". *Frontiers in Built Environment* 3 (cit. on p. 26).

Chatzis, M. N., E. N. Chatzi, and S. P. Triantafyllou (2017b). "A Discontinuous Extended Kalman Filter for non-smooth dynamic problems". *Mechanical Systems and Signal Processing* 92, pp. 13–29 (cit. on pp. 23, 26, 52, 70, 157).

Chen, Z., A. Malpani, P. Chalasani, A. Deguet, S. S. Vedula, P. Kazanzides, and R. H. Taylor (2016). "Virtual fixture assistance for needle passing and knot tying". *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2343–2350 (cit. on p. 11).

Chiang, L. S., P. S. Jay, P. Valdastri, A. Menciassi, and P. Dario (2009). "Tendon sheath analysis for estimation of distal end force and elongation". *2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 332–337 (cit. on p. 21).

Ciuti, G., K. Skonieczna-Żydecka, W. Marlicz, V. Iacovacci, H. Liu, D. Stoyanov, A. Arezzo, M. Chiurazzi, E. Toth, and H. Thorlacius (2020). "Frontiers of robotic colonoscopy: a comprehensive review of robotic colonoscopes and technologies". *Journal of clinical medicine* 9.6, p. 1648 (cit. on p. 5).

CMR surgical (2023). *Commercial website of Cambridge Medical Robotics (CMR surgical)*. URL: `https://cmrsurgical.com/` (visited on 02/25/2023) (cit. on p. 4).

Conti, F. and O. Khatib (2005). "Spanning large workspaces using small haptic devices". *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*. IEEE, pp. 183–188 (cit. on p. 115).

Cursi, F., W. Bai, E. M. Yeatman, and P. Kormushev (2022). "Model Learning With Backlash Compensation for a Tendon-Driven Surgical Robot". *IEEE Robotics and Automation Letters* 7.3, pp. 7958–7965 (cit. on p. 66).

Dallemagne, B. and J. Marescaux (2010). "The ANUBIS™ project". *Minimally Invasive Therapy & Allied Technologies* 19.5, pp. 257–261 (cit. on pp. 4, 5).

Darzi, A. and Y. Munz (2004). "The impact of minimally invasive surgical techniques". *Annu. Rev. Med.* 55, pp. 223–237 (cit. on p. 3).

De Donno, A., F. Nageotte, P. Zanne, L. Zorn, and M. d. Mathelin (2013). "Master/slave control of flexible instruments for minimally invasive surgery". *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 483–489 (cit. on pp. 62, 64).

De Geeter, J., H. Van Brussel, J. De Schutter, and M. Decreton (1997). "A smoothly constrained Kalman filter". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.10, pp. 1171–1177 (cit. on p. 82).

Dinh, B. K., L. Cappello, M. Xiloyannis, and L. Masia (2016). "Position control using adaptive backlash compensation for bowden cable transmission in soft wearable exoskeleton". *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5670–5676 (cit. on p. 22).

Do, T., T. Tjahjowidodo, M. Lau, T. Yamamoto, and S. Phee (2014). "Hysteresis modeling and position control of tendon-sheath mechanism in flexible endoscopic systems". *Mechatronics* 24.1, pp. 12–22 (cit. on pp. 8, 21).

Dominjon, L., J. Perret, and A. Lécuyer (2007). "Novel devices and interaction techniques for human-scale haptics". *The Visual Computer* 23.4, pp. 257–266 (cit. on p. 115).

Dragan, A. D. and S. S. Srinivasa (2013). "A policy-blending formalism for shared control". *The International Journal of Robotics Research* 32.7, pp. 790–805 (cit. on pp. 9, 10).

Enayati, N., E. D. Momi, and G. Ferrigno (2016). "Haptics in Robot-Assisted Surgery: Challenges and Benefits". *IEEE Reviews in Biomedical Engineering* 9, pp. 49–65 (cit. on p. 14).

Feng, J., X. Li, X. Xiao, B. Ouyang, and J. Fang (2021). "Virtual Fixtures Assistance for Safe Polyp Dissection in Minimally Invasive Robotic Surgery". *2021 5th International Conference on Automation, Control and Robots (ICACR)*, pp. 150–155 (cit. on p. 11).

Ferraguti, F., C. Secchi, and C. Fantuzzi (2013). "A tank-based approach to impedance control with variable stiffness". *2013 IEEE International Conference on Robotics and Automation*, pp. 4948–4953 (cit. on p. 121).

Fontanelli, G. A., G.-Z. Yang, and B. Siciliano (2018). "A comparison of assistive methods for suturing in MIRS". *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4389–4395 (cit. on p. 11).

Gama, J., I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia (2014). "A survey on concept drift adaptation". *ACM Computing Surveys* 46.4, pp. 1–37 (cit. on p. 131).

Gibo, T. L., L. N. Verner, D. D. Yuh, and A. M. Okamura (2009). "Design considerations and human-machine performance of moving virtual fixtures". *2009 IEEE International Conference on Robotics and Automation*, pp. 671–676 (cit. on p. 11).

Grasa, Ó. G., E. Bernal, S. Casado, I. Gil, and J. M. M. Montiel (2014). "Visual SLAM for Handheld Monocular Endoscope". *IEEE Transactions on Medical Imaging* 33.1, pp. 135–146 (cit. on pp. 42, 52).

Hagenow, M., E. Senft, R. Radwin, M. Gleicher, B. Mutlu, and M. Zinn (2021). "Corrective Shared Autonomy for Addressing Task Variability". *IEEE Robotics and Automation Letters* 6.2, pp. 3720–3727 (cit. on p. 75).

Haidegger, T. (2019). "Autonomy for Surgical Robots: Concepts and Paradigms". *IEEE Transactions on Medical Robotics and Bionics* 1.2, pp. 65–76 (cit. on p. 6).

Hannaford, B. and Jee-Hwan Ryu (2002). "Time-domain passivity control of haptic interfaces". *IEEE Transactions on Robotics and Automation* 18.1, pp. 1–10 (cit. on p. 127).

Hart, S. G. and L. E. Staveland (1988). "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research". *Advances in psychology*. Vol. 52. Elsevier, pp. 139–183 (cit. on p. 114).

Hersch, M., A. Billard, and S. Bergmann (2012). "Iterative Estimation of Rigid-Body Transformations". *Journal of Mathematical Imaging and Vision* 43.1, pp. 1–9 (cit. on p. 146).

Hogan, N. (1985). "Impedance Control: An Approach to Manipulation: Part I—Theory". *Journal of Dynamic Systems, Measurement, and Control* 107.1, pp. 1–7 (cit. on p. 74).

Horan, B. and S. Nahavandi (2008). "Intuitive haptic control surface for mobile robot motion control". *2008 IEEE International Workshop on Safety, Security and Rescue Robotics*. IEEE, pp. 121–127 (cit. on p. 115).

Intuitive Surgical (2022). *Commercial website of Intuitive Surgical Inc.* URL: `https://www.intuitive.com/en-us` (visited on 12/15/2022) (cit. on pp. 3–6).

Javdani, S., H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell (2018). "Shared autonomy via hindsight optimization for teleoperation and teaming". *The International Journal of Robotics Research* 37.7, pp. 717–742 (cit. on pp. 10, 76).

Johnson & Johnson (2023). *Commercial website of Johnson & Johnson.* URL: `https://www.jnj.com/` (visited on 02/06/2023) (cit. on p. 4).

Kaouk, J. H., R. K. Goel, G.-P. Haber, S. Crouzet, and R. J. Stein (2009). "Robotic single-port transumbilical surgery in humans: initial report". *BJU international* 103.3, pp. 366–369 (cit. on p. 4).

Karlsson, J., M. Anguelova, and M. Jirstrand (2012). "An Efficient Method for Structural Identifiability Analysis of Large Dynamic Systems*". *IFAC Proceedings Volumes* 45.16, pp. 941–946 (cit. on pp. 97, 139).

Kastritsi, T., F. Dimeas, and Z. Doulgeri (2018). "Progressive Automation with DMP Synchronization and Variable Stiffness Control". *IEEE Robotics and Automation Letters* 3.4, pp. 3789–3796 (cit. on p. 75).

Katzourakis, D., M. Alirezaei, J. C. F. de Winter, M. Corno, R. Happee, A. Ghaffari, and R. Kazemi (2011). "Shared control for road departure prevention". *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1037–1043 (cit. on p. 9).

Kilby, W., J. R. Dooley, G. Kuduvalli, S. Sayeh, and C. R. Maurer (2010). "The CyberKnife® Robotic Radiosurgery System in 2010". *Technology in Cancer Research & Treatment* 9.5, pp. 433–452 (cit. on p. 7).

Kim, H., M. Hwang, J. Kim, J. M. You, C.-S. Lim, and D.-S. Kwon (2020). "Effect of backlash hysteresis of surgical tool bending joints on task performance in teleoperated flexible endoscopic robot". *The International Journal of Medical Robotics and Computer Assisted Surgery* 16.1 (cit. on p. 8).

Ko, K. and T. Sakkalis (2014). "Orthogonal projection of points in CAD/CAM applications: an overview". *Journal of Computational Design and Engineering* 1.2, pp. 116–127 (cit. on p. 148).

Kobayashi, H. and H. Nakamura (1992). "A scaled teleoperation". *[1992] Proceedings IEEE International Workshop on Robot and Human Communication*. IEEE, pp. 269–274 (cit. on p. 115).

Kronander, K. and A. Billard (2016). "Stability Considerations for Variable Impedance Control". *IEEE Transactions on Robotics* 32.5, pp. 1298–1305 (cit. on p. 127).

Kühl, P., M. Diehl, T. Kraus, J. P. Schlöder, and H. G. Bock (2011). "A real-time algorithm for moving horizon state and parameter estimation". *Computers & chemical engineering* 35.1, pp. 71–83 (cit. on p. 127).

Leal Ghezzi, T. and O. Campos Corleta (2016). "30 Years of Robotic Surgery". *World Journal of Surgery* 40.10, pp. 2550–2557 (cit. on p. 8).

Lee, D.-H., M. Hwang, and D.-S. Kwon (2019). "Robotic endoscopy system (easyEndo) with a robotic arm mountable on a conventional endoscope". *2019 International Conference on Robotics and Automation (ICRA)*, pp. 367–372 (cit. on p. 4).

Lee, H., C. Ju, S. Park, S. Park, and H. I. Son (2017). "Preliminary user evaluation of inaccuracy in haptic guidance for teleoperated maintenance task of nuclear power plant". *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 467–469 (cit. on p. 14).

Liu, T. and M. C. Cavusoglu (2016). "Needle Grasp and Entry Port Selection for Automatic Execution of Suturing Tasks in Robotic Minimally Invasive Surgery". *IEEE Transactions on Automation Science and Engineering* 13.2, pp. 552–563 (cit. on p. 11).

Losey, D. P. and M. K. O'Malley (2018a). "Including Uncertainty when Learning from Human Corrections". *Conference on Robot Learning*, pp. 123–132. arXiv: 1806.02454 (cit. on p. 76).

Losey, D. P. and M. K. O'Malley (2019). "Learning the Correct Robot Trajectory in Real-Time from Physical Human Interactions". *ACM Transactions on Human-Robot Interaction* 9.1, pp. 1–19 (cit. on p. 76).

Losey, D. P. and M. K. O'Malley (2018b). "Trajectory Deformations From Physical Human–Robot Interaction". *IEEE Transactions on Robotics* 34.1, pp. 126–138 (cit. on p. 75).

Mahmoud, N., T. Collins, A. Hostettler, L. Soler, C. Doignon, and J. M. M. Montiel (2019). "Live Tracking and Dense Reconstruction for Handheld Monocular Endoscopy". *IEEE Transactions on Medical Imaging* 38.1, pp. 79–89 (cit. on pp. 12, 42).

Marinho, M. M., B. V. Adorno, K. Harada, and M. Mitsuishi (2019). "Dynamic Active Constraints for Surgical Robots Using Vector-Field Inequalities". *IEEE Transactions on Robotics* 35.5, pp. 1166–1185 (cit. on p. 11).

Martínez, N. and A. F. Villaverde (2020). "Nonlinear Observability Algorithms with Known and Unknown Inputs: Analysis and Implementation". *Mathematics* 8.11, p. 1876 (cit. on p. 139).

Masone, C., P. R. Giordano, H. H. Bülthoff, and A. Franchi (2014). "Semi-autonomous trajectory generation for mobile robots with integral haptic shared control". *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6468–6475 (cit. on p. 75).

Mayor, N., A. S. Coppola, and B. Challacombe (2022). "Past, present and future of surgical robotics". *Trends in Urology & Men's Health* 13.1, pp. 7–10 (cit. on p. 5).

Medtronic (2023). *Commercial website of Medtronic.* URL: https://www.medtronic.com (visited on 02/06/2023) (cit. on p. 4).

Meguenani, A., V. Padois, J. Da Silva, A. Hoarau, and P. Bidaud (2016). "Energy based control for safe human-robot physical interaction". *International Symposium on Experimental Robotics*. Springer, pp. 809–818 (cit. on p. 128).

Moccia, R., M. Selvaggio, L. Villani, B. Siciliano, and F. Ficuciello (2019). "Vision-based Virtual Fixtures Generation for Robotic-Assisted Polyp Dissection Procedures". *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7934–7939 (cit. on pp. 11, 15).

Mountney, P. and G.-Z. Yang (2010). "Motion compensated SLAM for image guided surgery". *International conference on medical image computing and computer-assisted intervention*. Springer, pp. 496–504 (cit. on p. 52).

Nageotte, F., P. Zanne, C. Doignon, and M. de Mathelin (2009). "Stitching Planning in Laparoscopic Surgery: Towards Robot-assisted Suturing". *The International Journal of Robotics Research* 28.10, pp. 1303–1321 (cit. on p. 11).

Nageotte, F., L. Zorn, P. Zanne, and M. De Mathelin (2020). "STRAS: A modular and flexible telemanipulated robotic device for intraluminal surgery". *Handbook of Robotic and Image-Guided Surgery*. Elsevier, pp. 123–146 (cit. on pp. 5, 32, 136).

Nelson, A. T. (2000). "Nonlinear estimation and modeling of noisy time series by dual Kalman filtering methods". PhD thesis. Oregon Graduate Institute of Science and Technology (cit. on p. 84).

Nelson, R. J., J. S. S. Chavali, N. Yerram, P. Babbar, and J. H. Kaouk (2017). "Current status of robotic single-port surgery". *Urology Annals* 9.3, pp. 217–222 (cit. on p. 5).

Niemeyer, G., C. Preusche, S. Stramigioli, and D. Lee (2016). "Telerobotics". *Springer handbook of robotics*. Springer, pp. 1085–1108 (cit. on pp. 9, 115).

Olivieri, E., G. Barresi, D. G. Caldwell, and L. S. Mattos (2018). "Haptic Feedback for Control and Active Constraints in Contactless Laser Surgery: Concept, Implementation, and Evaluation". *IEEE Transactions on Haptics* 11.2, pp. 241–254 (cit. on pp. 11, 15).

Oosterhout, J. v., J. G. W. Wildenbeest, H. Boessenkool, C. J. M. Heemskerk, M. R. d. Baar, F. C. T. v. d. Helm, and D. A. Abbink (2015). "Haptic Shared Control in Tele-Manipulation: Effects of Inaccuracies in Guidance on Task Execution". *IEEE Transactions on Haptics* 8.2, pp. 164–175 (cit. on pp. 14, 124).

Ortmaier, T., M. Groger, D. Boehm, V. Falk, and G. Hirzinger (2005). "Motion estimation in beating heart surgery". *IEEE Transactions on Biomedical Engineering* 52.10, pp. 1729–1740 (cit. on p. 10).

Osa, T., N. Sugita, and M. Mitsuishi (2018). "Online Trajectory Planning and Force Control for Automation of Surgical Tasks". *IEEE Transactions on Automation Science and Engineering* 15.2, pp. 675–691 (cit. on p. 11).

Ott, L., F. Nageotte, P. Zanne, and M. de Mathelin (2011). "Robotic Assistance to Flexible Endoscopy by Physiological-Motion Tracking". *IEEE Transactions on Robotics* 27.2, pp. 346–359 (cit. on pp. 10, 22, 37–39, 45, 49, 50, 52, 58, 67–71, 137).

Padoy, N. and G. D. Hager (2011). "Human-machine collaborative surgery using learned models". *2011 IEEE International Conference on Robotics and Automation*. IEEE, pp. 5285–5292 (cit. on p. 10).

Page, E. S. (1954). "Continuous inspection schemes". *Biometrika* 41.1, pp. 100–115 (cit. on p. 132).

Papageorgiou, D. and Z. Doulgeri (2020). "A control scheme for haptic inspection and partial modification of kinematic behaviors". *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 9752–9758 (cit. on pp. 125, 128).

Pegna, J. and F.-E. Wolter (1996). "Surface Curve Design by Orthogonal Projection of Space Curves Onto Free-Form Surfaces". *Journal of Mechanical Design* 118.1, pp. 45–52 (cit. on p. 148).

Peine, J. W., V. Agrawal, and W. J. Peine (2012). "Effect of backlash on surgical robotic task proficiency". *2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pp. 799–804 (cit. on p. 8).

Peter, S. D. S. and G. W. Holcomb III (2008). *History of minimally invasive surgery*. Philadelphia: Elsevier Health Sciences (cit. on p. 2).

Phee, S. J., A. P. Kencana, V. A. Huynh, Z. L. Sun, S. C. Low, K. Yang, D. Lomanto, and K. Y. Ho (2010). "Design of a master and slave transluminal endoscopic robot for natural orifice transluminal endoscopic surgery". *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 224.7, pp. 1495–1503 (cit. on p. 4).

Phee, S. J., N. Reddy, P. W. Chiu, P. Rebala, G. V. Rao, Z. Wang, Z. Sun, J. Y. Wong, and K.-Y. Ho (2012). "Robot-assisted endoscopic submucosal dissection is effective in treating patients with early-stage gastric neoplasia". *Clinical Gastroenterology and Hepatology* 10.10, pp. 1117–1121 (cit. on p. 5).

Pignat, E., J. Silvério, and S. Calinon (2022). "Learning from demonstration using products of experts: Applications to manipulation and task prioritization". *The International Journal of Robotics Research* 41.2, pp. 163–188 (cit. on p. 76).

Poignonec, T., F. Nageotte, and B. Bayle (2023a). "A fading memory discontinuous EKF for the online model identification of cable-driven robots with backlash". *2023 American Control Conference (ACC)*. Accepted. (cit. on pp. 16, 158).

Poignonec, T., F. Nageotte, N. Zemiti, and B. Bayle (2021). "Simultaneous haptic guidance and learning of task parameters during robotic teleoperation – a geometrical approach". *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3619–3625 (cit. on pp. 17, 125, 126, 134, 144, 147, 149, 160).

Poignonec, T., P. Zanne, B. Rosa, and F. Nageotte (2020). "Towards In Situ Backlash Estimation of Continuum Robots Using an Endoscopic Camera". *IEEE Robotics and Automation Letters* 5.3, pp. 4788–4795 (cit. on pp. 23, 26, 50, 134, 159).

Poignonec, T., P. Zanne, N. Zemiti, B. Bayle, and F. Nageotte (2023b). "Image-based backlash width identification for eye-in-hand robotized flexible endoscopes". *IEEE Robotics and Automation Letters*. Submitted for publication. (cit. on p. 17).

Prasad, S. M., S. M. Prasad, H. S. Maniar, C. Chu, R. B. Schuessler, and R. J. Damiano Jr (2004). "Surgical robotics: impact of motion scaling on task performance". *Journal of the American College of Surgeons* 199.6, pp. 863–868 (cit. on pp. 8, 115).

Pugin, F., P. Bucher, and P. Morel (2011). "History of robotic surgery : From AESOP® and ZEUS® to da Vinci®". *Journal of Visceral Surgery* 148.5 (cit. on p. 4).

Raiola, G., S. S. Restrepo, P. Chevalier, P. Rodriguez-Ayerbe, X. Lamy, S. Tliba, and F. Stulp (2018). "Co-manipulation with a library of virtual guiding fixtures". *Autonomous Robots* 42.5, pp. 1037–1051 (cit. on pp. 75, 128).

Rawlings, J. B. (2015). "Moving Horizon Estimation". *Encyclopedia of Systems and Control*. Ed. by J. Baillieul and T. Samad. London: Springer, pp. 799–804 (cit. on pp. 82, 127).

Reilink, R., S. Stramigioli, A. M. Kappers, and S. Misra (2011). "Evaluation of flexible endoscope steering using haptic guidance". *The International Journal of Medical Robotics and Computer Assisted Surgery* 7.2, pp. 178–186 (cit. on pp. 4, 11).

Reilink, R., S. Stramigioli, and S. Misra (2013a). "3D position estimation of flexible instruments: marker-less and marker-based methods". *International journal of computer assisted radiology and surgery* 8.3, pp. 407–417 (cit. on pp. 59, 61, 66).

Reilink, R., S. Stramigioli, and S. Misra (2013b). "Image-based hysteresis reduction for the control of flexible endoscopic instruments". *Mechatronics* 23.6, pp. 652–658 (cit. on pp. 9, 22, 23, 26, 35, 66–71).

Reiter, A., R. E. Goldman, A. Bajo, K. Iliopoulos, N. Simaan, and P. K. Allen (2011). "A learning algorithm for visual pose estimation of continuum robots". *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2390–2396 (cit. on p. 61).

Ren, J., R. V. Patel, K. A. McIsaac, G. Guiraudon, and T. M. Peters (2008). "Dynamic 3-D Virtual Fixtures for Minimally Invasive Beating Heart Procedures". *IEEE Transactions on Medical Imaging* 27.8, pp. 1061–1070 (cit. on p. 11).

Restrepo, S. S., G. Raiola, P. Chevalier, X. Lamy, and D. Sidobre (2017). "Iterative virtual guides programming for human-robot comanipulation". *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 219–226 (cit. on p. 75).

Richa, R., P. Poignet, and Chao Liu (2010). "Three-dimensional Motion Tracking for Beating Heart Surgery Using a Thin-plate Spline Deformable Model". *The International Journal of Robotics Research* 29.2, pp. 218–230 (cit. on p. 10).

Robotics Surgical (2022). *Commercial website of Robotics Surgical.* URL: `https://robotics-surgical.com/` (visited on 09/11/2022) (cit. on pp. 4, 5).

Rochdi, Y., F. Giri, J. B. Gning, and F. Z. Chaoui (2010). "Identification of block-oriented systems in the presence of nonparametric input nonlinearities of switch and backlash types". *Automatica* 46.5, pp. 864–877 (cit. on p. 55).

Rosa, B., V. Bordoux, and F. Nageotte (2019). "Combining differential kinematics and optical flow for automatic labeling of continuum robots in minimally invasive surgery". *Frontiers in Robotics and AI* 6, p. 86 (cit. on p. 61).

Rosenberg, L. B. (1993). "Virtual fixtures: Perceptual tools for telerobotic manipulation". *Proceedings of IEEE Virtual Reality Annual International Symposium*, pp. 76–82 (cit. on p. 10).

Rydén, F. and H. J. Chizeck (2012). "Forbidden-region virtual fixtures from streaming point clouds: Remotely touching and protecting a beating heart". *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3308–3313 (cit. on p. 11).

Saeidi, H., J. R. Wagner, and Y. Wang (2017). "A Mixed-Initiative Haptic Teleoperation Strategy for Mobile Robotic Systems Based on Bidirectional Computational Trust Analysis". *IEEE Transactions on Robotics* 33.6, pp. 1500–1507 (cit. on p. 9).

Schostek, S., M. O. Schurr, and G. F. Buess (2009). "Review on aspects of artificial tactile feedback in laparoscopic surgery". *Medical Engineering & Physics* 31.8, pp. 887–898 (cit. on p. 3).

Schulman, J., J. Ho, C. Lee, and P. Abbeel (2016). "Learning from Demonstrations Through the Use of Non-rigid Registration". *Robotics Research: The 16th International Symposium ISRR.* Ed. by M. Inaba and P. Corke. Springer Tracts in Advanced Robotics. Cham: Springer International Publishing, pp. 339–354 (cit. on p. 11).

Self, R., M. Harlan, and R. Kamalapurkar (2019). "Online inverse reinforcement learning for nonlinear systems". *2019 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 296–301 (cit. on pp. 76, 127).

Selvaggio, M., G. A. Fontanelli, F. Ficuciello, L. Villani, and B. Siciliano (2018). "Passive Virtual Fixtures Adaptation in Minimally Invasive Robotic Surgery". *IEEE Robotics and Automation Letters* 3.4, pp. 3129–3136 (cit. on p. 75).

Selvaggio, M., M. Cognetti, S. Nikolaidis, S. Ivaldi, and B. Siciliano (2021). "Autonomy in Physical Human-Robot Interaction: A Brief Survey". *IEEE Robotics and Automation Letters* 6.4, pp. 7989–7996 (cit. on p. 74).

Selvaggio, M., A. M. G. E, R. Moccia, F. Ficuciello, and B. Siciliano (2019). "Haptic-guided shared control for needle grasping optimization in minimally invasive robotic surgery". *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* Macau, China: IEEE, pp. 3617–3623 (cit. on p. 11).

Sestini, L., B. Rosa, E. De Momi, G. Ferrigno, and N. Padoy (2021). "A Kinematic Bottleneck Approach for Pose Regression of Flexible Surgical Instruments Directly From Images". *IEEE Robotics and Automation Letters* 6.2, pp. 2938–2945 (cit. on p. 35).

Simon, D. (2010). "Kalman Filtering with State Constraints: A Survey of Linear and Nonlinear Algorithms". *Control Theory & Applications, IET* 4, pp. 1303–1318 (cit. on pp. 27, 82–84, 124).

Smisek, J., M. M. van Paassen, and A. Schiele (2015). "Haptic guidance in bilateral teleoperation: Effects of guidance inaccuracy". *2015 IEEE World Haptics Conference (WHC)*, pp. 500–505 (cit. on pp. 14, 94).

Spaner, S. J. and G. L. Warnock (1997). "A Brief History of Endoscopy, Laparoscopy, and Laparoscopic Surgery". *Journal of Laparoendoscopic & Advanced Surgical Techniques* 7.6, pp. 369–373 (cit. on p. 2).

Stryker (2022). *Commercial website of Stryker*. URL: https://www.stryker.com/ (visited on 09/11/2022) (cit. on p. 7).

Sun, Z., Z. Wang, and S. J. Phee (2014). "Elongation Modeling and Compensation for the Flexible Tendon–Sheath System". *IEEE/ASME Transactions on Mechatronics* 19.4, pp. 1243–1250 (cit. on p. 21).

Swanström, L. L., Y. Khajanchee, and M. A. Abbas (2008). "Natural Orifice Transluminal Endoscopic Surgery: The Future of Gastrointestinal Surgery". *The Permanente Journal* 12.2, pp. 42–47 (cit. on p. 3).

Taylor, R., P. Jensen, L. Whitcomb, A. Barnes, R. Kumar, D. Stoianovici, P. Gupta, Z. Wang, E. Dejuan, and L. Kavoussi (1999). "A Steady-Hand Robotic System for Microsurgical Augmentation". *The International Journal of Robotics Research* 18.12, pp. 1201–1210 (cit. on p. 8).

Troccaz, J. (2013). *Medical robotics*. John Wiley & Sons (cit. on p. 12).

Vaiana, N., S. Sessa, F. Marmo, and L. Rosati (2018). "A class of uniaxial phenomenological models for simulating hysteretic phenomena in rate-independent mechanical systems and materials". *Nonlinear Dynamics* 93.3, pp. 1647–1669 (cit. on p. 71).

Villaverde, A. F., N. D. Evans, M. J. Chappell, and J. R. Banga (2019). "Input-Dependent Structural Identifiability of Nonlinear Systems". *IEEE Control Systems Letters* 3.2, pp. 272–277 (cit. on pp. 97, 139).

Vörös, J. (2010). "Modeling and identification of systems with backlash". *Automatica* 46.2, pp. 369–374 (cit. on p. 22).

Vörös, J. (2018). "Modelling and identification of nonlinear cascade systems with backlash input and static output nonlinearities". *Mathematical and Computer Modelling of Dynamical Systems* 24.6, pp. 593–609 (cit. on p. 55).

Wang, L., Z. Chen, P. Chalasani, R. M. Yasin, P. Kazanzides, R. H. Taylor, and N. Simaan (2017). "Force-controlled exploration for updating virtual fixture geometry in model-mediated telemanipulation". *Journal of Mechanisms and Robotics* 9.2 (cit. on pp. 11, 15).

Webster III, R. J. and B. A. Jones (2010). "Design and kinematic modeling of constant curvature continuum robots: A review". *The International Journal of Robotics Research* 29.13, pp. 1661–1683 (cit. on p. 31).

Weede, O., H. Mönnich, B. Müller, and H. Wörn (2011). "An intelligent and autonomous endoscopic guidance system for minimally invasive surgery". *2011 IEEE International Conference on Robotics and Automation*, pp. 5762–5768 (cit. on p. 9).

Wheeler, R. C. (1986). "Endoscopic apparatus". U.S. pat. 4607621A. W. A. Inc (cit. on p. 2).

Xia, J., D. Huang, Y. Li, and N. Qin (2020). "Iterative learning of human partner's desired trajectory for proactive human–robot collaboration". *International Journal of Intelligent Robotics and Applications* (cit. on p. 75).

Xiong, L., C. B. Chng, C. K. Chui, P. Yu, and Y. Li (2017). "Shared control of a medical robot with haptic guidance". *International Journal of Computer Assisted Radiology and Surgery* 12.1, pp. 137–147 (cit. on p. 11).

Xu, W., C. C. Y. Poon, Y. Yam, and P. W. Y. Chiu (2017). "Motion compensated controller for a tendon-sheath-driven flexible endoscopic robot". *The International Journal of Medical Robotics and Computer Assisted Surgery* 13.1 (cit. on p. 21).

Yang, G.-Z. et al. (2017). "Medical robotics—Regulatory, ethical, and legal considerations for increasing levels of autonomy". *Science Robotics* 2.4 (cit. on pp. 6, 9).

Yang, L., Y. Li, D. Huang, J. Xia, and X. Zhou (2022). "Spatial Iterative Learning Control for Robotic Path Learning". *IEEE Transactions on Cybernetics*, pp. 1–10 (cit. on p. 75).

Yasin, R., P. Chalasani, N. Zevallos, M. Shahbazi, Z. Li, A. Deguet, P. Kazanzides, H. Choset, R. H. Taylor, and N. Simaan (2021). "Evaluation of Hybrid Control and Palpation Assistance for Situational Awareness in Telemanipulated Task Execution". *IEEE Transactions on Medical Robotics and Bionics* 3.1, pp. 31–43 (cit. on p. 15).

Yip, M. C., J. A. Sganga, and D. B. Camarillo (2017). "Autonomous Control of Continuum Robot Manipulators for Complex Cardiac Ablation Tasks". *Journal of Medical Robotics Research* 02.1, p. 1750002 (cit. on pp. 6, 9).

Zeestraten, M. J. A., I. Havoutis, and S. Calinon (2018). "Programming by Demonstration for Shared Control With an Application in Teleoperation". *IEEE Robotics and Automation Letters* 3.3, pp. 1848–1855 (cit. on pp. 74, 124).

Zein, M. K., A. Sidaoui, D. Asmar, and I. H. Elhajj (2020). "Enhanced Teleoperation Using Autocomplete". *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9178–9184 (cit. on p. 75).

Zhan, J., J. Cartucho, and S. Giannarou (2020). "Autonomous Tissue Scanning under Free-Form Motion for Intraoperative Tissue Characterisation". *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11147–11154 (cit. on p. 11).

Zhang, Z., B. Rosa, O. Caravaca-Mora, P. Zanne, M. J. Gora, and F. Nageotte (2021). "Image-Guided Control of an Endoscopic Robot for OCT Path Scanning". *IEEE Robotics and Automation Letters* 6.3, pp. 5881–5888 (cit. on p. 11).

Zimmer Biomet (2022). *Commercial website of Zimmer Biomet*. URL: `https://www.zimmerbiomet.com/` (visited on 09/11/2022) (cit. on p. 7).

Zorn, L., F. Nageotte, P. Zanne, A. Legner, B. Dallemagne, J. Marescaux, and M. de Mathelin (2018). "A novel telemanipulated robotic assistant for surgical endoscopy: Preclinical application to esd". *IEEE Transactions on Biomedical Engineering* 65.4, pp. 797–808 (cit. on pp. 4, 5, 24, 25, 31, 32, 54, 60).