Abbas Tariverdi

Modeling and Control of a Continuum Robot for Medical Interventions

Dissertation for the Degree of PhD 2024
Department of Physics
Faculty of Mathematics and Natural Sciences ISSN 1501-7710 No. 2713

Abbas Tariverdi

Modeling and Control of a Continuum Robot for Medical Interventions

2024

UNIVERSITY OF OSLO

Faculty of Mathematics and Natural Sciences

Abbas Tariverdi

Modeling and Control of a Continuum Robot for Medical Interventions

UNIVERSITY
OF OSLO

Abbas Tariverdi

Modeling and Control of a Continuum Robot for Medical Interventions

Thesis submitted for the degree of Philosophiae Doctor

Department of Physics
Faculty of Mathematics and Natural Sciences

2024

"My Story Is A Lot Like Yours,
Only [Slightly] Interesting 'Cause It Involves [Math And] Robots."
Bender, Futurama (S04E11)

Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo
No. 2713

ISSN 1501-7710

Cover: UiO.
Print production: Graphic center, University of Oslo.

To my mother, a walking miracle.

Abstract

A challenge in vascular surgeries is the precise positioning of catheters. Robotic systems, especially flexible soft/serial-link robots, might be employed to assist surgeons during catheterizations by providing real-time sensing, better hand-eye coordination, and more reliable dexterity, especially in deep-seated regions within the human body. This dissertation aims to advance the modeling and control of catheterizations coupled with magnetic actuation. Mainly catheters/guidewires, i.e., Continuum Manipulators (CMs), are studied in this thesis.

To alleviate the physical burden of a clinician, methods of wireless actuation can be integrated with catheters or guidewires with advanced controllability that potentially will perform better than current actuation systems. By omitting the mechanical structures required for navigation of conventional robots, magnetic actuation wirelessly actuate robots, i.e., surgical tools such as catheters and guidewires, thus substantially reducing their size and complexity. This enables the surgical tools to manoeuvre better in confined areas inside biological bodies.

Although magnetically-actuated catheters/guidewires are structurally simpler than comparable devices, such magnetic tools exhibit complex mechanical behavior. This thesis focuses on how these dynamics can be analyzed in potentially clinically-relevant applications. The contributions are twofold and are presented in the form of two research questions. This dissertation suggests approaches to i) modeling the behavior of CMs and ii) devising control algorithms to enable their steering.

A novel approach is presented to derive models for complex mechanisms with large deformations that respects their intrinsic properties and preserves the structure of the configuration space. The experimental results show an average absolute error and maximum error in tip position estimation of 0.13 mm and 0.58 mm, respectively, for a manipulator length of 60.55 mm. A balance between computational bandwidths and prediction quality is established with a Neural Networks-driven model which can be used for real-time applications where the experiments suggest maximum and mean absolute errors of $(1.27°, 0.23°)$, $(4.69°, 1.27°)$, and $(0.60°, 0.10°)$ for rotations around the x, y, and z-axes in the tip, respectively, for a manipulator length of 95 mm. Furthermore, the dissertation explores the use of Reinforcement Learning techniques, Attractor Dynamics, and the Execution Extended Rapidly-Exploring Random Trees methods for an untethered magnetically actuated milliscale particle, and examines the performance of a simple optimal control in a guidewire steering scenario. The experimental results show an average success rate of 98.86% over 30 episodes demonstrating the viability of the proposed Reinforcement Learning method. After an introduction, we present a series of studies, divided into chapters, with each chapter concerning challenges related to a particular research question.

Sammendrag

En utfordring i vaskulærkirurgi er nøyaktig plassering av katetre. Robotiske systemer, spesielt fleksible myke/serie-koblede roboter, kan benyttes for å bistå kirurger under kateterisering ved å tilby sanntidsregistrering, bedre hånd-øye-koordinasjon, og økt fingerferdighet, særlig i dyptliggende områder inne i menneskekroppen. Denne avhandlingen sikter mot å fremme modelleringen og styringen av kateteriseringer koblet med magnetisk aktuering. Hovedfokus er på katetre/guideledninger, dvs. Kontinuum-Manipulatorer (KM). For å redusere den fysiske belastningen for en kliniker, kan metoder for trådløs aktuering integreres med katetre eller guideledninger med avansert kontrolleringsevne, som potensielt kan prestere bedre enn dagens aktiveringssystemer. Ved å fjerne de mekaniske strukturene nødvendige for navigering av konvensjonelle roboter, kan magnetisk aktuering trådløst aktivere roboter, altså kirurgiske verktøy som katetre og guideledninger, noe som vesentlig reduserer deres størrelse og kompleksitet. Dette gir de kirurgiske verktøyene bedre manøvreringsevne i trange områder inne i biologiske organismer.

Selv om magnetisk aktuerte katetre/guideledninger er strukturelt enklere enn sammenlignbare enheter, viser slike magnetiske verktøy kompleks mekanisk oppførsel. Denne avhandlingen fokuserer på hvordan disse dynamikkene kan analyseres i potensielt klinisk relevante applikasjoner. Bidragene er todelt og blir presentert i form av to forskningsspørsmål. Denne avhandlingen foreslår tilnærminger til i) modellering av oppførselen til KMs og ii) utforming av styringsalgoritmer for å muliggjøre deres styring.

En ny metode blir introdusert for å avlede modeller for komplekse mekanismer med store deformasjoner som tar hensyn til deres iboende egenskaper og bevarer strukturen i konfigurasjonsrommet. De eksperimentelle resultatene viser en gjennomsnittlig absolutt feil og maksimal feil i estimeringen av tuppens posisjon på henholdsvis 0,13 mm og 0,58 mm, med en manipulatorlengde på 60,55 mm. En balanse mellom beregningskapasitet og kvaliteten på prediksjonene er oppnådd ved hjelp av en modell basert på nevrale nettverk. Denne kan benyttes for sanntidsapplikasjoner hvor eksperimentene antyder maksimale og gjennomsnittlige absolutte feil på $(1,27°, 0,23°)$, $(4,69°, 1,27°)$, og $(0,60°, 0,10°)$ ved rotasjoner rundt henholdsvis x-, y-, og z-aksene i tuppen, for en manipulatorlengde på 95 mm. Videre utforsker avhandlingen bruken av teknikker innenfor forsterkningslæring, Attractor Dynamics og Execution Rapidly-Exploring Random Trees for en magnetisk drevet partikkel i millimeterskala. Den gransker også ytelsen av en enkel optimal kontroll i et guidewire-styringsscenario. De eksperimentelle resultatene viser en gjennomsnittlig suksessrate på 98,86% over 30 episoder, som bekrefter levedyktigheten til den foreslåtte metoden for forsterkningslæring.

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of *Philosophiae Doctor* at the University of Oslo. The research presented here was conducted at the University of Oslo, under the supervision of professors Ørjan G. Martinsen (Main Supervisor), Ole J. Elle, Jim Tørresen, and Kim Mathiassen (Co Supervisors).

The thesis is a collection of 5 papers, presented in a narrative order. The papers are preceded by 6 chapters providing introduction and background for the included papers that relates them to each other and provides background information and motivation for the work. The papers are joint work with other authors. I am the corresponding author of the papers in which I am the first author.

Acknowledgements

I kindly appreciate my supervisors Ørjan G. Martinsen, Jim Tørresen, Ole J. Elle, and Kim Mathiassen for their guidance, novel suggestions and feedback to my thesis. I truly thank Ørjan once again for his kind, friendly, and continually encouraging attitude that not just made this collaboration a pleasant and memorable experience, but also profoundly influenced my growth as a scholar and an individual. Being his student will forever be a treasured chapter in my life's journey. Also, I thank all the Bioimpedance, ROBIN, and Intervention Center graduate students for their friendship.

I express my deepest appreciation to my parents and sisters, who are the greatest blessings that I have been gifted. Without their encouragement, never-ending love, and prayers, I would never have been able to conduct this research and write this thesis.

Abbas Tariverdi
Oslo, January 2024

List of Papers

Paper I

Abbas Tariverdi, Venkatasubramanian Kalpathy Venkiteswaran, Ørjan Grøttem Martinsen, Ole Jacob Elle, Jim Tørresen, Sarthak Misra "Dynamic Modeling of Soft Continuum Manipulators Using Lie Group Variational Integration". In: *PLOS ONE*. Vol. 15, Issue. 7 (2020), pp. e0236121. DOI: 10.1371/journal.pone.0236121.

Paper II

Abbas Tariverdi, Venkatasubramanian Kalpathy Venkiteswaran, Michiel Richter, Ole J. Elle, Jim Tørresen, Kim Mathiassen, Sarthak Misra and Ørjan G. Martinsen "A Recurrent Neural-Network-Based Real-Time Dynamic Model for Soft Continuum Manipulators". In: *Frontiers in Robotics and AI*. Vol. 8, (2021), pp. 631303. DOI: 10.3389/frobt.2021.631303.

Paper III

Abbas Tariverdi, Ulysse Cote-Allard, Kim Mathiassen, Ole J. Elle, Håvard Kalvøy, Ørjan G. Martinsen, Jim Tørresen "Reinforcement Learning-based Switching Controller for a Milliscale Robot in a Constrained Environment". Accepted in *IEEE Transactions on Automation Science and Engineering*. Early access, available at DOI: 10.1109/TASE.2023.3259905.

Paper IV

Seyed MohammadReza Sajadi, **Abbas Tariverdi**, Henrik Brun, Ole J. Elle, Kim Mathiassen "Novel Robotic-Assisted Transesophageal Echocardiography System - Design, Prototyping, and Deep Learning Modeling". Submitted in *IEEE Transactions on Robotics*.

Paper V

Abbas Tariverdi, Vegard Søyseth, Ørjan G. Martinsen, Håvard Kalvøy, Kim Mathiassen, Ole J. Elle, Jim Tørresen, and Mats Erling Høvin "Physics-Based Simulation and Control Framework for Steering a Magnetically-Actuated Guidewire". In: *8th International Conference on Control, Deci-*

sion and Information Technologies (CoDIT). Vol. 1, (2022), pp.770–774. DOI: 10.1109/CoDIT55151.2022.9803963.

Contents

Abstract iii

Sammendrag v

Preface vii

List of Papers ix

Contents xi

List of Figures xv

List of Tables xix

1 Introduction 1
- 1.1 Robotics in MIS . . . 1
- 1.2 Actuation Mechanisms for CMs . . . 2
- 1.3 Modeling and Control of CMs . . . 3
- 1.4 Research Objectives and Outline of the Thesis . . . 4
- 1.5 Scientific Output: Summary of Papers . . . 6
- References . . . 9

2 Background 11
- 2.1 Models for CMs . . . 11
- 2.2 Control Methods for CMs . . . 14
- References . . . 16

3 Dynamic Models for CMs 21
- 3.1 Exact Nonlinear Dynamic Model for Soft CMs: Lie Group Variational Integration . . . 21
- 3.2 Real-time Dynamic Model of CMs: RNN-based Prediction 25
- 3.3 A Rapid Physics-base Kinematic Model . . . 35
- 3.4 Findings and Implications . . . 38
- References . . . 41

4 Control Strategies for CMs 43
- 4.1 RL-based Switching Controller for an Untethered Magnetic Particle . . . 44
- 4.2 Optimal Kinematic-based Controller for Magnetic Guidewire 61
- 4.3 Findings and Implications . . . 63

References 64

5 Discussion 67
References 71

6 Concluding Remarks and Future Prospects 73
6.1 Conclusions 73
6.2 Future Work 75

Papers 78

I Dynamic Modeling of Soft Continuum Manipulators Using Lie Group Variational Integration 79
I.1 Introduction 80
I.2 Continuum Manipulator Dynamics 83
I.3 Lie Group Variational Integrators for the Forced Continuum Manipulator 86
I.4 Simulation and Experimental Results 94
I.5 Discussion 104
I.6 Conclusions and Future Work 110
I.A Preliminaries on Lie groups and Lie algebra 111
References 111

II A Recurrent Neural-Network-Based Real-Time Dynamic Model for Soft Continuum Manipulators 117
II.1 Introduction 118
II.2 Problem Statement 120
II.3 Proposed RNN-based Model 121
II.4 Simulations Results 122
II.5 Experimental Results 131
II.6 Discussion 142
II.7 Conclusion 143
II.A Cosserat Rod Theory 143
References 144

III Reinforcement Learning-based Switching Controller for a Milliscale Robot in a Constrained Environment 149
III.1 Introduction 150
III.2 Problem Statement 154
III.3 Preliminaries 155
III.4 Methods: Reinforcement Learning, Attractor Dynamics, and ERRT-based Switching Controllers 157
III.5 Simulations, Experiments, and Results 167
III.6 Discussions 179
III.7 Conclusion 180
References 181

IV Novel Robotic-Assisted Transesophageal Echocardiography System - Design, Prototyping, and Deep Learning Modeling **187**
IV.1 Introduction . . . 188
IV.2 Mechanical Design and Prototyping . . . 190
IV.3 Kinematic Model . . . 198
IV.4 Experiments . . . 200
IV.5 Discussion . . . 207
IV.6 Conclusion . . . 211
References . . . 211

V Physics-Based Simulation and Control Framework for Steering a Magnetically-Actuated Guidewire **215**
V.1 Introduction . . . 216
V.2 Problem Statement and Motivation . . . 217
V.3 Method . . . 218
V.4 Simulation Results . . . 222
V.5 Discussion . . . 222
V.6 Conclusion . . . 224
References . . . 224

Appendices **227**

A Some Mathematical Concepts with Physical Interpretations **229**
References . . . 232

List of Figures

1.1 Thesis conceptual diagram 6
1.2 Overview of papers grouped by the research questions 7

2.1 Overview of models for CMs 12

3.1 Chapter 3 overview 21
3.2 Steps for deriving CM discrete dynamics 23
3.3 Configuration of the nodes and filters 23
3.4 Magnetic field generation setup 24
3.5 A soft manipulator with discretization nodes 27
3.6 RNN-based model 28
3.7 Experimental setup consists of stationary electromagnets 30
3.8 Initial and time-evolved configurations and applied wrenches . . 31
3.9 Robotic-assisted Transesophageal Echocardiography system . . . 33
3.10 Terminology for TEE manual manipulation 33
3.11 Conceptual design of robotic TEE system 34
3.12 TEE tip positions 35
3.13 Dipoles force interaction 37
3.14 Dipoles torque interaction 37
3.15 External magnet rotating around y-axis 37
3.16 A guidewire partitioned into multiple segments 38

4.1 Chapter 4 overview 43
4.2 RL-based switching controller architecture 47
4.3 Control-loop mechanism when Sub-controllers are activated . . . 48
4.4 Observations for training Sub-controller 2 48
4.5 Attractor Dynamics-based switching controller architecture . . . 49
4.6 ERRT-based switching controller architecture 52
4.7 Inputs and outputs for the obstacle processing unit 53
4.8 Experimental setup for RL- and ERRT-based control strategies . 55
4.9 Online obstacle avoidance of UR5 carrying an undisturbed particle 57
4.10 Online obstacle avoidance of UR5 carrying a disturbed particle . 58
4.11 Initial environment with boundary obstacles and initial path planning by robot in ERRT approach 58
4.12 Initial, target, and disturbance locations, explored area and selected path 59
4.13 Set-point regulation with three disturbances in one episode . . . 60
4.14 Proposed steering setup including an external magnet 62
4.15 Projected force applied to the guidewire tip dipole 63

4.16 Maximum projected force applied to the guidewire tip dipole . . 64

I.1 Initial and time-evolved configurations of the continuum manipulator 84
I.2 Steps for deriving continuum manipulator discrete dynamics . . 87
I.3 Configuration of the nodes of the model and distributed filters . 92
I.4 Block diagram of the proposed prediction filter 93
I.5 AlSi05 rod configurations, in-plane experiment 96
I.6 AlSi05 rod, in-plane experiment results 97
I.7 AlSi05 rod, out-of-plane experiment results 98
I.8 PDMS rod without magnet 99
I.9 PDMS rod, planar motion results 101
I.10 PDMS rod with embedded magnet at tip in 2D experiment . . . 102
I.11 PDMS rod with embedded magnet at tip, planar experiment results 103
I.12 Magnetic field generation setup 104
I.13 Scene reconstruction for circular motion of PDMS rod with embedded tip magnet 106
I.14 PDMS rod with two embedded magnets, 2D experiment 107
I.15 PDMS rod with two embedded magnets, 2D experiment results 108
I.16 High and low frequency simulation comparison 110

II.1 A soft manipulator with discretization nodes 121
II.2 Recurrent Neural Network-based model 123
II.3 First simulation: model architecture, rod properties, simulation parameters, and prediction and measured results 126
II.4 Second simulation: model architecture, rod properties and simulation parameters 128
II.5 Second simulation prediction and measured results 129
II.6 Evaluation example of second simulation: prediction and measured results 130
II.7 Third simulation: model architecture, rod properties and simulation parameters 132
II.8 Third simulation prediction and measured results 133
II.9 Experimental setup consists of stationary electromagnets 135
II.10 Initial and time-evolved configurations and applied wrenches . . 137
II.11 Model architecture used for the experiment 138
II.12 Experiment results 139
II.13 Computation time for BVP solutions varies with error tolerance and node count 141
II.14 Computation bandwidth and RMS error of prediction 141

III.1 RL-based switching controller architecture 158
III.2 Control scheme when Sub-controller 1 is activated 159
III.3 Singularity-free workspace of the UR 159
III.4 Control scheme when Sub-controller 2 is activated 160
III.5 Switching controller architecture 161
III.6 Observations to train Sub-controller 2 162

III.7 ERRT-based switching controller architecture 165
III.8 Inputs and outputs for the obstacle processing unit 166
III.9 Experimental setup including UR5, a permanent magnet, and RGB camera 168
III.10 Block and sphere Neodymium magnets 169
III.11 Simulation environment used to train Rainbow controller 170
III.12 Mean and standard deviation of loss and rewards 170
III.13 Set-point regulation without disturbances in an episode 171
III.14 Set-point regulation with disturbances in an episode 172
III.15 Online obstacle avoidance of UR5 carrying an undisturbed particle 174
III.16 Online obstacle avoidance of UR5 carrying a disturbed particle . 175
III.17 Initial environment with boundary obstacles and initial path planning by robot in ERRT approach 176
III.18 Initial, target, and disturbance locations, explored area and selected path 177
III.19 Set-point regulation with three disturbances in one episode . . . 178

IV.1 Terminology for TEE manual manipulation 191
IV.2 Conceptual design of the proposed robotic system 193
IV.3 Advance/Withdraw and Turn to Left/Right DoFs in Add-on Robotic subsystem 194
IV.4 Advance/Withdraw and Turn to Left/Right DoFs in Robotic Holder subsystem 194
IV.5 Flex to Left/Right and Handle-Inclination DoFs in Robotic Holder subsystem 195
IV.6 Anteflex/Retroflex, Flex to Left/Right, and Rotate Forward/Backward DoFs in Robotic Holder subsystem 196
IV.7 Electrical schematic of the robotic system 197
IV.8 Robotic-assisted Transesophageal Echocardiography system . . . 198
IV.9 Proposed RNN-based kinematic model of the developed robotic system 201
IV.10 RNN-based kinematic model 201
IV.11 TEE experimental setup 202
IV.12 Data-sets positions for different gastroscope tube shapes 204
IV.13 Density distribution along the x-axis 205
IV.14 Density distribution along the y-axis 205
IV.15 Density distribution along the z-axis 205
IV.16 Poses for a square-like trajectory 206
IV.17 0-degree data-set: measured data, prediction, and RMS error . . 208
IV.18 45-degrees data-set: measured data, prediction, and RMS error . 209
IV.19 90-degrees data-set: measured data, prediction, and RMS error . 210

V.1 Beam bends due to an attraction field 218
V.2 Beam deflects due to the misalignment of two dipoles 219
V.3 Guidewire is partitioned into multiple segments and joints . . . 219
V.4 Proposed steering setup including an external magnet 221

V.5 Rotating external magnet modeled as 2000 dipole, resulting flux, and the applied force 221
V.6 Projected force applied to tip dipole alongside GI tract 223
V.7 Maximum projected force applied to guidewire tip dipole 223

List of Tables

2.1	Computationally-heavy dynamic approaches for CMs	12
2.2	Fast dynamic methods for CMs	13
2.3	Fast kinematic approaches for CMs	13
2.4	Controllers for CMs/small-scale robots	15
3.1	Training and validation times for proposed RNN models	32
I.1	References on dynamics/static analysis of CMs	81
I.2	Parameters definition in Lagrangian equation	85
I.3	Simulation parameters for in-plane experiment of AlSi05 rod	95
I.4	Simulation parameters for out-of-plane experiment of AlSi05 rod	97
I.5	Simulation parameters for in-plane experiment of PDMS rod	100
I.6	Simulation parameters for in-plane and circular-motion experiments of PDMS rod (with tip magnet)	100
I.7	Simulation parameters for in-plane experiment of square cross-section PDMS rod (with 2 magnets)	105
I.8	Maximum and mean absolute errors for AlSi05 and PDMS rods	109
II.1	Maximum and mean absolute errors for distal and proximal nodes	140
IV.1	Motion range based on clinical requirements	192
IV.2	Actuators technical properties	198
IV.3	Properties of 0-, 45-, and 90-degree-bend data-sets	204
V.1	Static simulations of continuum manipulators	217

Chapter 1

Introduction

Minimally Invasive Surgery (MIS) has become a standard methodology in many clinical interventions over the past few decades due to its clear advantages for operational success and patient recovery. The benefits include reduced trauma during procedures, less pain, decreased blood loss, shorter hospital stays, quicker recovery time, and lower risk of infection.

1.1 Robotics in MIS

Surgical robotics has significantly grown over the past decade to enable robotic systems in various complex medical procedures that are arguably difficult to perform with conventional means. Robotic systems are used to augment and extend the capabilities of surgeons, offering outstanding levels of dexterity and precision in diagnosis and treatment. Robotic methods are well equipped for performing multiple tasks simultaneously, e.g., handling multiple tools and 3D visualizations.

Better robot designs, modeling, and control methods are needed to meet the current challenges in MIS. Reachability, a high level of dexterity, and large elastic deformability are the primary driving factors behind the growth of research in the design, modeling, and control of Continuum Manipulators (CMs), which are flexible and deformable robots composed of soft and elastic materials and can serve as possible substitutes for rigid robots. Flexible CMs have recently generated interest in several fields, especially in minimally invasive surgical robotics and interventional medicine, such as catheter-based endovascular intervention [Bur+13; Gra+00] and cardiac surgeries [CP09; KH11]. In contrast to conventional rigid link manipulators, soft manipulators are able to reshape their configurations to allow for redundancies in path planning and are capable of precise and delicate manipulation of objects in complex and varying environments.

1.1.1 Potential Applications of CMs in Medical Interventions

Continuum manipulators, with their ability to change shape and position flexibly along their entire length, are particularly well-suited for minimally invasive procedures. They are adept at navigating through body lumens, natural orifices, or small surgical incisions, which makes them ideal for a range of applications in MIS [Dup+22]:

These manipulators are highly advantageous in navigating the vascular system for targeted interventions due to their slenderness and dexterity, offering minimal invasiveness and precision. In cardiac surgeries, continuum robots can perform delicate procedures within the heart, minimizing tissue damage and improving

patient outcomes. Their narrow curvilinear shape allows them to be effectively used in surgeries like laryngeal procedures and other interventions requiring access to confined areas.

1.1.2 Design Requirements for CMs in MIS

Designing continuum manipulators for MIS involves considerations of flexibility, precision, biocompatibility, and more. Key requirements include dexterity, elastic deformability, precision in control, and miniaturization for patient safety and reduced trauma [RDM19; Wei+23].

1.1.3 Expected Performances of CMs

The technical contributions of continuum manipulators in MIS are expected to enhance surgical precision, improve operational efficiency, reduce patient recovery time, and show adaptability to various surgical scenarios, thus improving surgical procedures and patient care in MIS [Zha+23].

1.2 Actuation Mechanisms for CMs

Steerable manipulators, i.e., catheters in the context of MIS, are actuation-dependent devices, and there are numerous candidate actuation mechanisms for CMs, such as electric and mechanic actuations [Cia+13; WJ10; Yun+12]. Compared to these actuation mechanisms, magnetic actuation benefits from high dexterity, versatility, and wireless actuation [HSM18; Sik+19]. However, mechanically actuated catheters are the largest group in the literature. Mechanic actuation describes the most commercially available group of recent catheter actuation technologies, based on cable or tendon-actuated steering. It should be noted that in this work, both magnetic and mechanical actuations were investigated.

Magnetic fields can manipulate objects without any contacts. For CMs, this enables manipulation without the need for mechanical guidewires or other internal structures. Magnetic CMs consist of a passive elastic body and magnetic sections, typically permanent magnets or small electromagnetic coils. An externally generated magnetic field harmlessly permeates the patient's body and applies forces and torques on these magnets, causing the overall continuum structure to deform. Omitting the structures necessary for mechanical actuation allows magnetic catheters and endoscopes to be more flexible and thinner than tendon-wire designs, enabling access to difficult-to-reach areas. Magnetic CMs can either serve as interventional devices equipped with the necessary tools or guide other non-magnetic instruments. Magnetic actuation typically relies on alterations in a magnetic field, either internal or external, applied to the distal end of the catheter equipped with electromagnetic elements. The resulting magnetic force, with its specific magnitude and direction, is determined by the properties of these electromagnetic elements and the surrounding magnetic field. By controlling the

strength and orientation of the field, the magnetically active catheter tip can be guided inside the body.

Magnetically actuated catheters are found to be moving from the experimental scene toward the medical bench, with several magnetically steerable catheters already commercially available, for example, the Niobe and the Genesis (Stereotaxis Inc., MO, USA) [Arm+07; Tsu+06], the CGCI system (Magnetecs Corporation, CA, USA) [Moy+13]. Additionally, catheter navigation in the MRI is another growing field of research due to the potential of combining visualization techniques with steering [Lal+15].

Regarding the mechanic actuation, two subcategories can be distinguished regarding the tip mechanism in these catheters: 1) single segment actuation and 2) multiple segment actuation. The former group ensures deflection by using steering cables together with a bendable catheter tip made from a compliant material. The latter group provides deflection by the relative movement between separate elements or joints as a result of cable push and pull.

1.3 Modeling and Control of CMs

One of the key challenges in the field of CMs is developing a model for analyzing the dynamics of the manipulators and accurately predicting and controlling their behavior. Control, trajectory planning, and optimal design purposes of CMs typically rely on dynamic models of manipulators, especially in unknown and unstructured environments such as inside the human body. Existing models focus primarily on static or quasi-static approaches to make the computations tractable. Such models usually do not account for the geometric nonlinearities (e.g., large deformations) and do not respect the conservation of dynamical properties of the system (e.g., energy and momentum maps conservation) or the structures of configuration space. On the other hand, dynamical modeling approaches, contain dynamics of CMs and also take into account time-varying responses of manipulators, including high-frequency modes which may also preserve the invariants (energy, momentum maps) or the symplectic structure of CMs, especially for long-time simulations.

Accurately steering CMs for clinically relevant tasks within a constrained environment requires proper control and/or pathfinding strategies, especially in the presence of disturbances such as cardiac and respiratory motions, the movement of the GI tract, and the bloodstream. It is obvious that the complexity of the intervascular structure and its disturbances are barriers to using offline-programmed paths, where an exact knowledge of the path from the insertion point to the target location should be known in advance. Therefore, developing appropriate control or pathfinding methods requires knowledge about the shape of CMs or a model to describe the behavior of CMs.

1.4 Research Objectives and Outline of the Thesis

Although CMs, especially in the field of MIS, provide surgeons with unique clinical possibilities, such as the capability of navigating through confined areas in biological bodies, their fabrication and development are subjected to a variety of challenges. The challenges are varied, from the design of CMs to the understanding and handling of their behavior.

Two key research questions have been identified by a vast literature search done in the papers and summarized in Chapter 2 as the primary driving force behind the research done in this thesis. Ultimately, the following were the investigated research questions and the thesis objectives:

RQ.1 CM Modeling: How to develop dynamic models of functionalized CMs and make the models more practically-relevant for novel applications?

In many tasks, dynamic models of CMs are crucial for control, trajectory planning, and optimal design purposes, especially in MIS for operations in unknown and unstructured environments such as inside biological bodies. In the literature, the primary focus is on static or quasi-static, non-real-time, or geometrically inexact approaches. Investigation of previous work reveals the need for a comprehensive modeling method for offline study of CMs' workspace, long-term simulations, and animation purposes. In this work, we derive different models with unique properties, from a systematic, actuation-independent, computationally intensive, geometrically exact model to a neural networks-driven model that establishes a balance between computational bandwidths and prediction quality.

RQ.2 CM Control: How to enable (vision-based) control and path finding strategies applied to an (un)tethered magnetic milliscale robot?

Due to the elastic characteristics and geometric nonlinearities (i.e., bending, torsion, shear, and elongation, including large deformation) of CMs, their dynamics may show highly nonlinear behavior and a complex workspace. Therefore, CMs require a closed-loop control system in real-time robotic applications. Conventional control and path-planning techniques rely on analytical models. Also, traditional controllers and online path planners are problem-specific or computationally heavy approaches, and their design for time-varying workspaces and uncertain environments (e.g., with obstacles and disturbances) becomes challenging. On the other hand, deep Reinforcement Learning (RL) methods are powerful algorithms that handle trajectory planning tasks and can be generalized for various dynamic workspaces in the presence of complex obstacles and uncertainties.

Remark 1.4.1. The research in this thesis focuses on developing dynamic models of CMs and control strategies for magnetic milliscale robots, with the aim of

advancing medical interventions in MIS. The performance metrics that guide this research are multifaceted, reflecting both the complexity and novelty of the applications.

For CM Modeling (**RQ.1**), the performance metrics include:

- Modeling Accuracy and Dynamics Analysis: In evaluating the performance of dynamic models for CMs, a key metric encompasses both Estimation and Modeling Accuracy. This metric focuses on the precision of model predictions, emphasizing the fidelity in capturing geometric nonlinearities, such as large deformations, and conserving the dynamic properties of the system. Precision is quantified through criteria like average absolute error and maximum error in tip position estimation for manipulators, measured in millimeters.
- Computational Bandwidth: A critical aspect of dynamic modeling in real-time applications is the balance between computational resource allocation and the quality of predictions. This is particularly pertinent for Neural Networks-driven models used in CMs. The focus is on achieving high prediction accuracy with minimal computational delay, ensuring that the models can be effectively integrated into real-time applications.
- Model Validation: In this thesis, validation encompasses comparing our developed dynamic models against outcomes from real-world experiments. This approach benchmarks the models' practical effectiveness, confirms their theoretical soundness, and ensures their applicability in real-life scenarios.

For CM Control (**RQ.2**), the performance metrics are:

- Success Rate: Demonstrating the viability of the Reinforcement Learning method and its real-life applicability through the average success rate over multiple episodes.
- Robustness to Disturbances: This metric gauges the control system's resilience against external perturbations, focusing on its consistency in achieving predefined objectives under variable conditions.
- Path Planning Efficiency: Benchmarking the RL-based algorithm against classical approaches like Attractor Dynamics and Execution Extended Rapidly-Exploring Random Trees to assess the system's efficiency.
- Control and Path finding Efficacy: In constrained environments, the effectiveness of control strategies and path finding techniques is paramount, including accuracy in trajectory planning and adaptability to changes in the environment.
- Handling Nonlinear Behavior and Complex Workspace: Evaluating the CM's ability to navigate complex workspace and its performance in real-time robotic application.

Additionally, the thesis addresses the practical implications of these metrics, aiming to reduce the physical burden on clinicians and improve surgical outcomes. The overarching goal is to enhance the dexterity and maneuverability of surgical tools within the human body, thereby potentially improving patient safety and surgical precision.

The thesis consists of this introduction chapter and five more chapters where we address background in Chapter 2, methods and experimental results addressing the research questions, discussion, and concluding remarks through Chapter 3 to Chapter 6. Chapter 3 gives an account of the development process of dynamic modeling algorithms to capture the governing dynamics of magnetic CMs and revisits the derivation and validation processes of real-time models for CMs. This chapter covers the research question **RQ.1** through Paper I, Paper II, Paper IV, and Paper V. The research question **RQ.2** is investigated in Chapter 4 which mainly concerns the control of CMs through Paper III, and Paper V. Lastly, discussions and conclusions are presented in Chapter 5 and Chapter 6. The papers come next, and then Appendix A provides physical interpretations of a few mathematical concepts used throughout the thesis. Figure 1.1 depicts the conceptual diagram for the thesis.

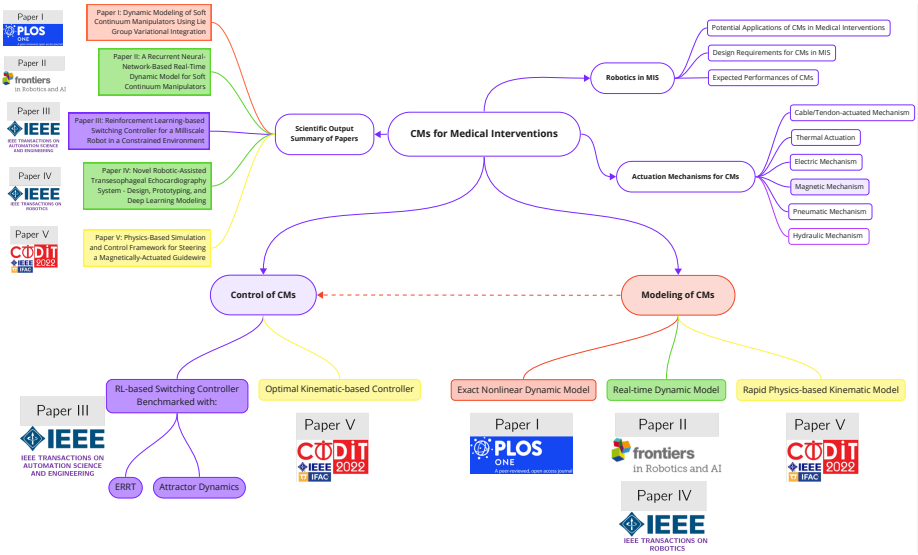


Figure 1.1: Conceptual diagram of the thesis.

1.5 Scientific Output: Summary of Papers

The studies conducted in the course of this thesis have been organized in a series of publications. Figure 1.2 shows how the papers are connected to the research

questions. Across all chapters, the content of this dissertation relates to the context set by the following scientific articles.

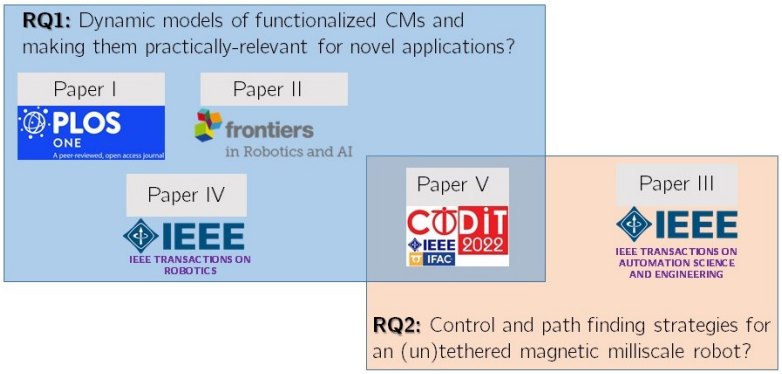


Figure 1.2: All the papers included in the thesis, grouped by the research questions they address. See Section 1.4 for the full research questions. Paper IV is submitted at the time of thesis submission.

Paper I focuses on the derivation and experimental validation of a dynamic algorithm for modeling and estimation of soft CMs' full spatial dynamics using Lie group variational integration. Using magnetic actuation, dynamic and static experiments were conducted on manipulators with rigid and soft materials (e.g., aluminum and Polydimethylsiloxane (PDMS)) to illustrate the validity of the presented algorithm for a wide range of experiments. This article's novelty can be stated as follows:

- The primary contribution of this article is the derivation and experimental validation of a dynamic model for forced continuum manipulators with soft materials undergoing spatial deformation. The model accounts for the nonlinearities of the continuum manipulator, including loading resulting from magnetic fields, gravity, and internal and external dissipation forces generated by friction.
- Due to the difficulty in obtaining knowledge about the internal and external dissipation forces, distributed estimation filters have been designed to take these forces into account and capture their behavior.

Paper II introduces and validates a real-time prediction framework based on a neural network approach for soft continuum manipulators. The proposed RNN-based parallel predictive scheme does not rely on computationally intensive algorithms; therefore, it is useful in real-time applications. The comparison results of the proposed model and a well-known model for continuum manipulators, i.e., Cosserat rod theory, are also provided, revealing the practical effectiveness of the proposed model. The contributions of this paper are as follows:

- Real-time prediction of soft manipulators' full spatial dynamics is considered in the proposed RNN-based algorithm by incorporating multiple lightweight RNN-based models.
- In traditional modeling approaches, there are no systematic methods to obtain knowledge about dissipation forces, in particular friction, during the modeling procedure. The presented algorithm intrinsically takes the dissipation forces into account and incorporates their effects into the model.
- Through an experiment, the results of the proposed RNN-based model and Cosserat rod theory method are compared, revealing the practical effectiveness of the proposed methodology.

Paper III presents a customized reinforcement learning-based magnetic manipulation and switching control mechanism to autonomously carry a particle (i.e., a milliscale robot) around obstacles within a constrained environment in the presence of disturbances. Applying wrenches to ferromagnetic particles is an essential part of magnetic actuation mechanisms. As a first step, only a single untethered magnetic object is considered in this paper, and the algorithm can be potentially used as an underlying control technique for magnetic CMs. The main contributions to this work are as follows:

- Implementing an RL-based switching framework for dynamic control of a particle in the presence of obstacles in two dimensions with visual feedback and using a robotic arm in a real-world scenario.
- Through simulations and experiments, the results of the proposed RL-based algorithm and classical approaches such as Attractor Dynamics, and Execution Extended Rapidly-Exploring Random Trees (ERRT) are compared, revealing the practical efficacy of the proposed methodology.

Paper IV highlights the development and design of a robotic-assisted transesophageal echocardiography (TEE) system. This work introduces a novel approach in robotizing TEE probes, along with a focus on real-time kinematics modeling using a RNN. The RNN model adeptly predicts the TEE transducer's position and orientation, addressing challenges like the probe's dead-zone and nonlinearities. These contributions represent a significant leap in robotic control and precision for TEE applications in medical interventions.

Paper V establishes a physics-based simulation framework for steering a magnetically actuated guidewire based on the linear elasticity and dipole theories. In the presented framework, a simplified integration scheme based on the finite-volume method is employed to model guidewire using the linear elasticity theory and forces resulting from the interference of magnetic fields to provide a rapid model reconstruction.

This study contributes by presenting a rapid quasi-static modeling technique for the simulation of soft manipulator control and formation. This article proposes a rapid multiphysics simulation framework for magnetically actuated continuum manipulators within an environment similar to the GI tract for closed-loop control applications to reduce the reality gap. The suggested technique is unique in that it combines rapid quasi-static models with soft guidewires controlled by magnetic fields, which may be employed in closed-loop control systems for precise navigation.

This chapter has established a foundational understanding of CMs, highlighting this thesis' core aspects. The next chapter will build on this foundation, exploring the historical development of the field, key studies, and theoretical frameworks. Our focus will be on identifying emerging challenges and potential advancements within this area of research.

References

[Arm+07] Armacost, M. P. et al. "Accurate and reproducible target navigation with the Stereotaxis Niobe® magnetic navigation system". In: *Journal of Cardiovascular Electrophysiology* vol. 18 (2007), S26–S31.

[Bur+13] Burgner, J. et al. "Debulking from within: a robotic steerable cannula for intracerebral hemorrhage evacuation". In: *IEEE Transactions on Biomedical Engineering* vol. 60, no. 9 (2013), pp. 2567–2575.

[Cia+13] Cianchetti, M. et al. "STIFF-FLOP surgical manipulator: Mechanical design and experimental characterization of the single module". In: *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2013, pp. 3576–3581.

[CP09] Carpi, F. and Pappone, C. "Stereotaxis Niobe® magnetic navigation system for endocardial catheter ablation and gastrointestinal capsule endoscopy". In: *Expert review of medical devices* vol. 6, no. 5 (2009), pp. 487–498.

[Dup+22] Dupont, P. E. et al. "Continuum robots for medical interventions". In: *Proceedings of the IEEE* vol. 110, no. 7 (2022), pp. 847–870.

[Gra+00] Grady, M. S. et al. "Experimental study of the magnetic stereotaxis system for catheter manipulation within the brain". In: *Journal of neurosurgery* vol. 93, no. 2 (2000), pp. 282–288.

[HSM18] Heunis, C., Sikorski, J., and Misra, S. "Flexible instruments for endovascular interventions: Improved magnetic steering, actuation, and image-guided surgical instruments". In: *IEEE robotics & automation magazine* vol. 25, no. 3 (2018), pp. 71–82.

[KH11] Kesner, S. B. and Howe, R. D. "Position control of motion compensation cardiac catheters". In: *IEEE Transactions on Robotics* vol. 27, no. 6 (2011), pp. 1045–1055.

[Lal+15] Lalande, V. et al. “In vivo demonstration of magnetic guidewire steerability in a MRI system with additional gradient coils”. In: *Medical physics* vol. 42, no. 2 (2015), pp. 969–976.

[Moy+13] Moya, A. et al. “Innovations in Heart Rhythm Disturbances: Cardiac Electrophysiology, Arrhythmias, and Cardiac Pacing”. In: *Revista Espanola de Cardiologia (English Edition)* vol. 66, no. 2 (2013), pp. 116–123.

[RDM19] Runciman, M., Darzi, A., and Mylonas, G. P. “Soft robotics in minimally invasive surgery”. In: *Soft robotics* vol. 6, no. 4 (2019), pp. 423–443.

[Sik+19] Sikorski, J. et al. “The ARMM system: An optimized mobile electromagnetic coil for non-linear actuation of flexible surgical instruments”. In: *IEEE transactions on magnetics* vol. 55, no. 9 (2019), pp. 1–9.

[Tsu+06] Tsuchida, K. et al. “Guidewire navigation in coronary artery stenoses using a novel magnetic navigation system: first clinical experience”. In: *Catheterization and cardiovascular interventions* vol. 67, no. 3 (2006), pp. 356–363.

[Wei+23] Wei, X. et al. “Modeling and control of cable-driven continuum robot used for minimally invasive surgery”. In: *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine* vol. 237, no. 1 (2023), pp. 35–48.

[WJ10] Webster III, R. J. and Jones, B. A. “Design and kinematic modeling of constant curvature continuum robots: A review”. In: *The International Journal of Robotics Research* vol. 29, no. 13 (2010), pp. 1661–1683.

[Yun+12] Yun, C.-H. et al. “Multi-degree-of-freedom ultrasonic micromotor for guidewire and catheter navigation: The NeuroGlide actuator”. In: *Applied Physics Letters* vol. 100, no. 16 (2012), p. 164101.

[Zha+23] Zhang, T. et al. “Design and Modeling of Multi-vertebrae Continuum Robot for Minimally Invasive Surgery”. In: *Procedia Computer Science* vol. 226 (2023), pp. 85–91.

Chapter 2

Background

Continuum robots and soft robots represent two innovative approaches in robotic design, each with unique features suited for specific applications. While they share some similarities in flexibility and dexterity, their differences are crucial in the context of robotics[Li+22; WLK21].

Continuum robots, inspired by natural structures such as elephant trunks and tentacles, exhibit a continuous, curvilinear structure devoid of discrete joints, enabling smooth and natural movements. These robots are typically constructed from materials that balance rigidity and flexibility, like segmented metal or plastic, facilitating precise control necessary for complex medical procedures [RD99; WJ10]. In contrast, soft robots are designed with highly flexible, compliant materials like elastomers and bio-inspired substances, granting them remarkable adaptability and safety due to their compliance [Maj14; Tri+08]. These materials enable soft robots to mimic biological organisms, focusing on seamless integration with the human body and natural environments.

The actuation methods employed by these robots are pivotal to their functionality. Continuum robots often use tendon-driven systems, pneumatic or hydraulic actuation, and sometimes shape memory alloys, providing precise control and repeatable positioning, which are crucial for MIS applications [RS12]. Soft robots, however, utilize soft actuators such as pneumatic artificial muscles, electroactive polymers, or fluidic elastomer actuators. These actuators allow for adaptive and biomimetic movement but often at the expense of reduced precision [Li+22; Mad+04; Shi+16].

Modeling and control of soft Continuum Manipulators (CMs) have been a hotspot of study in recent years, with the overarching goal of developing accurate models and efficient control strategies that can handle the inherent complexity of these systems. This chapter presents an overview of the background and state-of-the-art in modeling and control of continuum manipulators, with a particular emphasis on the key developments, advances, and challenges in this field.

2.1 Models for CMs

Controlling and understanding almost all physical systems typically requires simulations of their dynamic and kinematic behavior over a wide range of time and space scales. In the literature, numerous approximate models have been investigated with the primary goal of striking a balance between computational complexity and precision. In other words, the resulting models trade off accuracy for computational efficiency. To get the big picture on modeling for CMs, Figure 2.1 depicts an overview of different modeling methods categorized based

on computation cost and the nature of the model (i.e., kinematic versus dynamic). Some articles on computationally-intensive dynamic models are cited in Table 2.1 and a few references on rapid dynamic and kinematic analyses for CMs are brought up in Table 2.2 and Table 2.3, respectively.

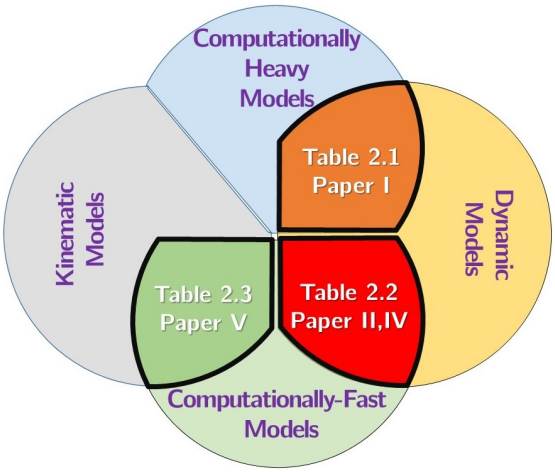


Figure 2.1: Overview of references grouped based on dynamic/kinematic and computationally heavy/fast methods. The intersections represent models that share attributes of the adjoining categories. This visualization also indicates the positioning of our research within this framework

Table 2.1: References on computationally-heavy dynamic methods for CMs

Ref.	**Modeling Approach and Properties**	**Robot Type/Application**
[AKN04]	Multi-body dynamics with bending energies	Guidewire modeling
[RB13; RB14]	High-fidelity lumped parameter model	Multisegment rod-driven continuum robots
[Ren+14]	Rigorous Cosserat geometrically exact model	CM prototype inspired by the octopus arm
[Dem+15; GDS19]	Variational Lie group formulation	General continuum manipulator

Following the references, the limitations of computationally-heavy dynamic methods for CMs discussed in Table 2.1 can be:

- Computational Resources: These methods often require significant computational power and time, which can limit their use in real-time

Table 2.2: References on fast dynamic analysis for CMs

Ref.	Modeling Approach and Properties	Robot Type/Application
[Dur+06]	3D beam theory with incremental approach	General catheter for interventional radiology
[TAR19]	Cosserat/Kirchhoff PDEs	Extensible rods, tendons, and fluidic chambers
[Wen+12]	Cosserat rod model and 3D elesticity	Guidewire/ Interventional Radiology procedures
[Ruc+10; WRC09]	Beam mechanics based on elastic energy (includes both bending and torsion)	Concentric tubes/ General MIS
[Tun13]	FEM with large deformation and inflation	Simulations on general medical robots

Table 2.3: References on fast kinematic analysis for CMs

Ref.	Modeling Approach and Properties	Robot Type/Application
[JW06]	Configuration space limited by actuator length	Backbone continuum robots
[GG19]	Cosserat rod approach	pneumatic continuum arm
[Che+21]	Integration of the spatial curve by a modal method	1-DoF pneumatic bending robot
[RW11]	Curve-based description	Concentric-tube robot

applications or for simulations that involve complex scenarios or long time horizons.

- Model Fidelity: While they may capture the dynamics of the system with high fidelity, the complexity of the models may introduce challenges in parameter estimation and validation.
- Scalability: These methods may not scale well as the complexity of the manipulator increases, such as when adding more segments or when considering interactions with complex environments.
- Specificity: The models and their validation might be highly tailored to specific applications or types of continuum manipulators or actuation mechanisms, which can limit their generalizability.

Fast dynamic analysis for CMs discussed in Table 2.2 may show the following shortcomings:

- Approximations: To achieve faster computations, these methods may make approximations that can reduce the accuracy of the models, particularly in predicting the behavior of the system under extreme conditions or interactions with the environment.

- Complex Dynamics and Validation: Ensuring the validity of the model across a wide range of operating conditions can be challenging when the methods prioritize computational speed. Simplifications necessary for speed can overlook complex dynamic phenomena such as nonlinear elastic effects, hysteresis, or dynamic interactions with fluids.

- Real-Time Constraints: Even though these methods are faster, there may still be constraints that prevent their use in high-frequency real-time control loops, especially when high precision is required.

Fast kinematic approaches discussed in Table 2.3 may have the following limitations:

- Dynamic Ignorance: These methods typically ignore the dynamics of the system, which can lead to discrepancies between the predicted and actual behavior, especially when the manipulator is subject to dynamic loading or when inertia plays a significant role.

- Simplifications: Kinematic models may oversimplify the physical interactions, such as friction, backlash, or material compressibility, which can be significant in some applications.

- Control Limitations: While fast for planning and simulation, kinematic models may not provide sufficient information for advanced control strategies that require dynamic information.

- Assumptions: They often rely on assumptions about the manipulator's shape or environment that may not hold in all situations, potentially reducing their applicability to novel or unstructured scenarios.

2.2 Control Methods for CMs

Control and path planning of CMs are broad research areas in which several types of controllers have been explored in the literature, ranging from model-based versus model-free, kinematic versus dynamic, and classical versus learning-based controllers. In addition, the control and localization of small-scale robots/particles are of utmost significance, where they essentially serve as actuation points for CMs, particularly magnetic CMs. Table 2.4 is a synopsis of some recent, pivotal articles on control and path planning in CMs. It is important to note that classical control methods developed using kinematic models or learning approaches taught on the static or kinematic behavior of CMs

Table 2.4: References on control algorithms for CMs or small-scale robots

Ref.	**Control Approach and Properties**	**Robot Type/Application**
[Kaz+22]	Dynamic model-based, classical adaptive sliding mode position control	Soft robotic arm
[Li+17]	Model-free kinematic adaptive Kalman filter-based position estimator	Pneumatic muscled CM
[Ber+20]	Kinematic model-based distal supervised learning method	Tendon driven robot
[Thu+18]	Dynamic model-based reinforcement learning control	Pneumatically actuated soft manipulator
[Jia+21; You+17]	Model-free kinematic reinforcement learning control	Honeycomb pneumatic manipulator
[Kha+02]	Model-based PID and Adaptive position controllers	3-DoF microrobot
[Xu+15]	Kinematic model|vision-based orientation|position proportional controller	Helical microswimmer

may not be accurate or efficient when CMs move fast enough (high-frequency movements), or in other words, display complex dynamic motion.

Each referenced control approach has its own set of limitations and potential areas for improvement, which are crucial to acknowledge for advancing the state of the art. Here is a discussion on possible limitations of the methods presented in Table 2.4:

- Dynamic Model-Based, Classical Adaptive Sliding Mode Position Control [Kaz+22]: This approach may struggle with system uncertainties and external disturbances. Adaptive sliding mode control is effective in handling uncertainties, but it may not be robust enough in highly dynamic environments. There's also the issue of chattering, a common problem in sliding mode control, which can cause excessive wear and tear on the robotic components.

- Model-Free Kinematic Adaptive Kalman Filter-Based Position Estimator [Li+17]: While model-free approaches have the advantage of not requiring precise physical models, they often lack the ability to predict system dynamics accurately, especially in complex environments. The adaptive Kalman filter can improve estimation, but it might be slow to adapt to sudden changes in dynamics or in the presence of non-Gaussian noise.

- Kinematic Model-Based Distal Supervised Learning Method [Ber+20]: These methods rely heavily on the accuracy of the kinematic models. If the model does not accurately represent the real system's dynamics, the control performance can degrade. Additionally, supervised learning methods require extensive data for training, and they may not generalize well to situations not encountered during training.
- Dynamic Model-Based Reinforcement Learning Control [Thu+18]: The success of reinforcement learning (RL) depends significantly on the quality of the reward function and the training environment. RL can be sample inefficient and may require a large amount of data to learn effective policies. Furthermore, RL-based controllers might exhibit unpredictable behavior in scenarios not encountered during training.
- Model-Free Kinematic Reinforcement Learning Control [Jia+21; You+17]: Similar to other RL approaches, these methods can suffer from poor generalization to new scenarios. Being model-free, they might also lack the ability to accurately predict the effects of control actions on system dynamics, which can lead to suboptimal control strategies.
- Model-Based PID and Adaptive Position Controllers [Kha+02]: PID controllers are relatively simple and may not handle complex system dynamics effectively. They are also prone to performance degradation in the presence of nonlinearities and external disturbances. Adaptive elements can improve performance, but they require careful tuning and may not adapt quickly to rapid changes in the system.
- Kinematic Model|Vision-Based Orientation|Position Proportional Controller [Xu+15]: This approach depends on the accuracy of both the kinematic model and the vision system. Misalignments or errors in the vision system can lead to poor control performance. Proportional controllers may also struggle with stability and precision in highly dynamic environments.

In the following chapter, we will discuss three novel algorithms that have been presented to address the three main types of modeling that we have discussed up until this point. These algorithms will cover computationally intensive dynamic models as well as fast dynamic and kinematic analyses.

References

[AKN04] Alderliesten, T., Konings, M. K., and Niessen, W. J. "Simulation of minimally invasive vascular interventions for training purposes". In: *Computer Aided Surgery* vol. 9, no. 1-2 (2004), pp. 3–15.

[Ber+20] Bern, J. M. et al. "Soft robot control with a learned differentiable model". In: *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE. 2020, pp. 417–423.

[Che+21] Chen, Y. et al. “Modal-based kinematics and contact detection of soft robots”. In: *Soft Robotics* vol. 8, no. 3 (2021), pp. 298–309.

[Dem+15] Demoures, F. et al. “Discrete variational Lie group formulation of geometrically exact beam dynamics”. en. In: *Numerische Mathematik* vol. 130, no. 1 (May 2015), pp. 73–123.

[Dur+06] Duriez, C. et al. “New approaches to catheter navigation for interventional radiology simulation”. In: *Computer aided surgery* vol. 11, no. 6 (2006), pp. 300–308.

[GDS19] Grazioso, S., Di Gironimo, G., and Siciliano, B. “A geometrically exact model for soft continuum robots: The finite element deformation space formulation”. In: *Soft robotics* vol. 6, no. 6 (2019), pp. 790–811.

[GG19] Gilbert, H. B. and Godage, I. S. “Validation of an extensible rod model for soft continuum manipulators”. In: *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE. 2019, pp. 711–716.

[Jia+21] Jiang, H. et al. “Hierarchical control of soft manipulators towards unstructured interactions”. In: *The International Journal of Robotics Research* vol. 40, no. 1 (2021), pp. 411–434.

[JW06] Jones, B. A. and Walker, I. D. “Practical kinematics for real-time implementation of continuum robots”. In: *IEEE Transactions on Robotics* vol. 22, no. 6 (2006), pp. 1087–1099.

[Kaz+22] Kazemipour, A. et al. “Adaptive dynamic sliding mode control of soft continuum manipulators”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 3259–3265.

[Kha+02] Khamesee, M. B. et al. “Design and control of a microrobotic system using magnetic levitation”. In: *IEEE/ASME transactions on mechatronics* vol. 7, no. 1 (2002), pp. 1–14.

[Li+17] Li, M. et al. “Model-free control for continuum robots based on an adaptive Kalman filter”. In: *IEEE/ASME Transactions on Mechatronics* vol. 23, no. 1 (2017), pp. 286–297.

[Li+22] Li, M. et al. “Soft actuators for real-world applications”. In: *Nature Reviews Materials* vol. 7, no. 3 (2022), pp. 235–249.

[Mad+04] Madden, J. D. et al. “Artificial muscle technology: physical principles and naval prospects”. In: *IEEE Journal of oceanic engineering* vol. 29, no. 3 (2004), pp. 706–728.

[Maj14] Majidi, C. “Soft robotics: a perspective—current trends and prospects for the future”. In: *Soft robotics* vol. 1, no. 1 (2014), pp. 5–11.

[RB13] Rone, W. S. and Ben-Tzvi, P. “Continuum robot dynamics utilizing the principle of virtual power”. In: *IEEE Transactions on Robotics* vol. 30, no. 1 (2013), pp. 275–287.

[RB14] Rone, W. S. and Ben-Tzvi, P. “Mechanics modeling of multisegment rod-driven continuum robots”. In: *Journal of Mechanisms and Robotics* vol. 6, no. 4 (2014), p. 041006.

[RD99] Robinson, G. and Davies, J. B. C. “Continuum robots-a state of the art”. In: *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)*. Vol. 4. IEEE. 1999, pp. 2849–2854.

[Ren+14] Renda, F. et al. “Dynamic model of a multibending soft robot arm driven by cables”. In: *IEEE Transactions on Robotics* vol. 30, no. 5 (2014), pp. 1109–1122.

[RS12] Rolf, M. and Steil, J. J. “Constant curvature continuum kinematics as fast approximate model for the Bionic Handling Assistant”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 3440–3446.

[Ruc+10] Rucker, D. C. et al. “Equilibrium conformations of concentric-tube continuum robots”. en. In: *The International Journal of Robotics Research* vol. 29, no. 10 (Sept. 2010), pp. 1263–1280.

[RW11] Rucker, D. C. and Webster, R. J. “Computing jacobians and compliance matrices for externally loaded continuum robots”. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 945–950.

[Shi+16] Shintake, J. et al. “Versatile soft grippers with intrinsic electroadhesion based on multifunctional polymer actuators”. In: *Advanced materials* vol. 28, no. 2 (2016), pp. 231–238.

[TAR19] Till, J., Aloi, V., and Rucker, C. “Real-time dynamics of soft and continuum robots based on Cosserat rod models”. In: *The International Journal of Robotics Research* vol. 38, no. 6 (2019), pp. 723–746.

[Thu+18] Thuruthel, T. G. et al. “Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators”. In: *IEEE Transactions on Robotics* vol. 35, no. 1 (2018), pp. 124–134.

[Tri+08] Trivedi, D. et al. “Soft robotics: Biological inspiration, state of the art, and future research”. In: *Applied bionics and biomechanics* vol. 5, no. 3 (2008), pp. 99–117.

[Tun13] Tunay, I. “Spatial continuum models of rods undergoing large deformation and inflation”. In: *IEEE Transactions on Robotics* vol. 29, no. 2 (Apr. 2013), pp. 297–307.

[Wen+12] Wen Tang et al. “A stable and real-time nonlinear elastic approach to simulating guidewire and catheter insertions based on cosserat rod”. In: *IEEE Transactions on Biomedical Engineering* vol. 59, no. 8 (Aug. 2012), pp. 2211–2218.

[WJ10] Webster III, R. J. and Jones, B. A. “Design and kinematic modeling of constant curvature continuum robots: A review”. In: *The International Journal of Robotics Research* vol. 29, no. 13 (2010), pp. 1661–1683.

[WLK21] Wang, X., Li, Y., and Kwok, K.-W. “A survey for machine learning-based control of continuum robots”. In: *Frontiers in Robotics and AI* vol. 8 (2021), p. 730330.

[WRC09] Webster, R., Romano, J., and Cowan, N. “Mechanics of precurved-tube continuum robots”. In: *IEEE Transactions on Robotics* vol. 25, no. 1 (Feb. 2009), pp. 67–78.

[Xu+15] Xu, T. et al. “Planar path following of 3-D steering scaled-up helical microswimmers”. In: *IEEE Transactions on Robotics* vol. 31, no. 1 (2015), pp. 117–127.

[You+17] You, X. et al. “Model-free control for soft manipulators based on reinforcement learning”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 2909–2915.

Chapter 3

Dynamic Models for CMs

This chapter deals with the research question **RQ.1** described in Section 1.4. Section 3.1 concerns Paper I (a computationally heavy dynamic model), Section 3.2 deals with Paper II and Paper IV (a computationally fast dynamic model), and the modeling part of Paper V is discussed in Section 3.3 (a rapid kinematic model reconstruction). An overview of Chapter 3 is shown in Figure 3.1.

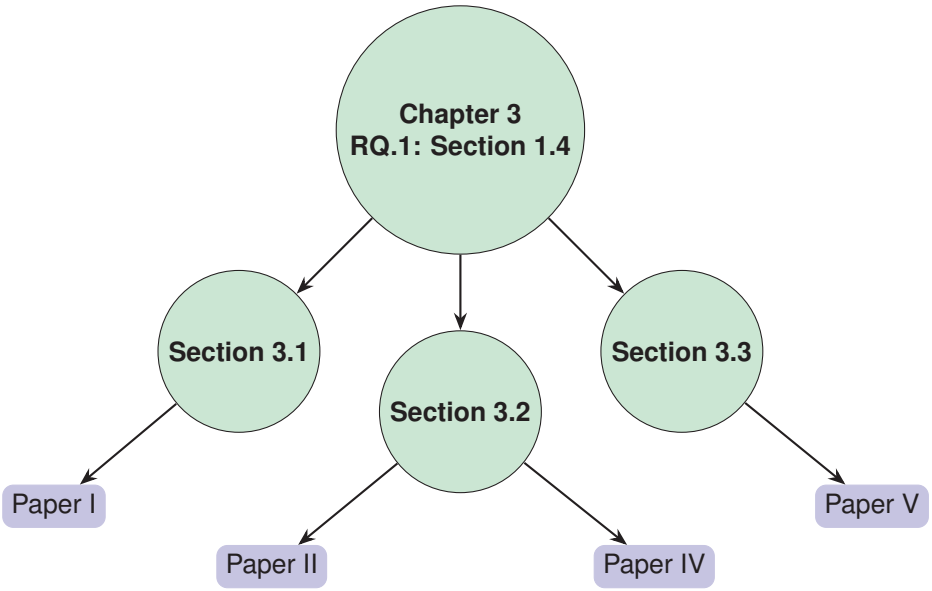


Figure 3.1: An overview of Chapter 3.

3.1 Exact Nonlinear Dynamic Model for Soft CMs: Lie Group Variational Integration

This section mainly deals with Paper I with regards to developing a Lie group variational solver for soft Continuum Manipulators (CMs). It is worth mentioning that the term solver refers to a numerical algorithm or computational method that solves the underlying equations governing the system's behavior. Finite element and difference methods have been essential computational tools for studying the dynamics and statics of many physical systems, including CMs. The main problem with these methods is that they discretize (partial) differential dynamic

equations to obtain motion-related solutions that do not necessarily satisfy the fundamental properties of continuum mechanical models. Although the stable and convergent finite solvers preserve the invariant geometric and topological structures underlying the continuous model such as symplectic structures, these characteristics can sometimes be lost in the computation process. For example, finite element methods do not necessarily preserve moments for systems with symmetry. In other words, an obtained solution for a free rigid body may change momentum and thereby fail to preserve invariant geometric structures [Dem+15; MR13]. It has been shown that this problem can be avoided by the use of Lie group variational integrators[BS99; Dem+15; LOL14].

Lagrange-d'Alembert's variational principle governs a wide variety of forced mechanical systems, including CMs. The object investigated in this section, is the exact nonlinear model of dissipative or forced CMs [Sim85], where the configuration space is a Lie group. Lie group variational integrators are derived for a given classical discrete Lagrangian, i.e., kinetic minus potential and conservative energies for CMs, by forming a discrete version of the Lagrange-d'Alembert principle. Combining variational integrators with Lie-group techniques enables integration schemes to respect not only the structure of the dynamics and its properties but also the structure of the configuration space.

Unlike finite element methods, the fundamental idea underlying the employed Lie group discrete variational integration is to discretize the variational principle rather than discretize the equations of motion. This leads to the fact that for non-dissipative and non-forced problems, variational integrators are symplectic and momentum conserving [Hai+06].

To sum up, it is worth emphasizing that Lie group variational integration schemes have not yet been fully explored, implemented, and tested in the real-world in the context of magnetically-actuated soft CMs modeling, and this section aims to contribute mainly in this direction. The primary contribution is the derivation and experimental validation of a dynamic model for forced CMs with soft materials undergoing spatial deformation. The model accounts for the nonlinearities of the continuum manipulator; loading results from magnetic fields, gravity, and internal and external dissipation forces generated by friction. Also, due to the difficulty in obtaining knowledge about the internal and external dissipation forces, distributed estimation filters have been designed to take these forces into account and capture their behavior.

3.1.1 Continuum Manipulator Dynamics and Kinematics

Firstly, we consider the kinematics of CMs undergoing large deflections; afterwards, we introduce a classical Lagrangian, i.e., kinetic minus potential and conservative energies. The continuous Euler-Lagrange equations, i.e., dynamics of CMs, are obtained by applying the Lagrange-d'Alembert principle to the action function associated with the Lagrangian (Section I.2 in Paper I).

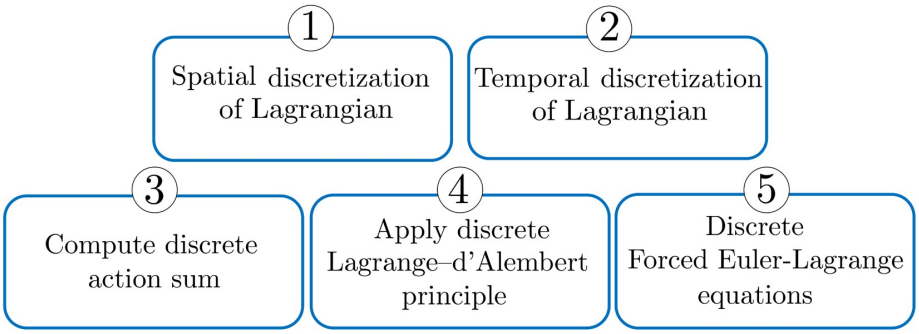


Figure 3.2: Steps 1 through 5 toward deriving CM discrete dynamics.

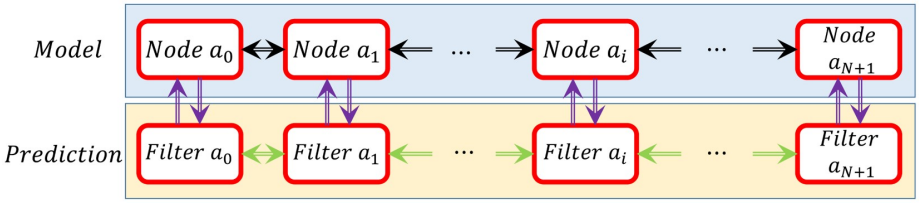


Figure 3.3: Configuration of the nodes of the model and the corresponding distributed filters. Filter a_i and Node a_i are coupled with the adjacent nodes in succession.

3.1.2 Lie Group Variational Integrators for the Forced CMs: Modeling and Estimation

In this subsection, we discuss deriving the discretized version of the forced Euler-Lagrange with conservative (e.g., gravity) and non-conservative forces (e.g., friction and external loads inserted by actuators). First, one needs to consider the spatial discretization of Lagrangian introduced earlier. Afterward, discrete Lagrange-d'Alembert equations need to be expressed on the configuration space, which is a Lie group. Figure 3.2 depicts the modeling procedure. By following the mentioned steps, discrete Euler-Lagrange equations for rotations and translations are derived (Section I.3.1 in Paper I).

It is discussed how the online distributed estimation algorithm [Lu95] predicts the model dissipation errors. The structure of the estimator mimics the structure of the model, i.e., discrete Euler-Lagrange equations in a distributed multi-systems configuration. We consider each node as an individual system coupled with the other adjacent nodes, i.e., neighbors, in succession. In other words, each node exchanges its local pose (position and orientation) with its neighbors. It should be noted that the estimation filters are designed and implemented for each node. Figure 3.3 shows the configuration of the distributed filters and nodes (Section I.3.2 in Paper I).

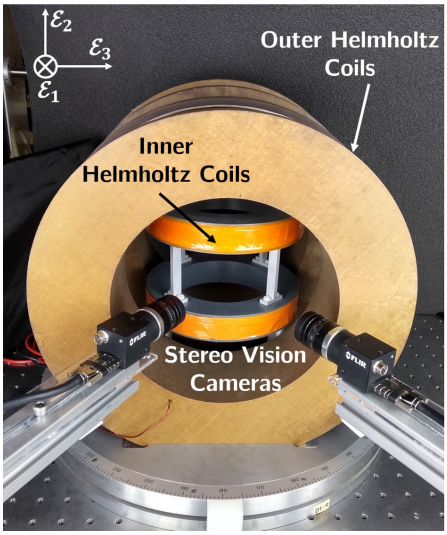


Figure 3.4: Magnetic field generation setup with the stereo vision cameras. Two nested pairs of Helmholtz coils generate uniform magnetic fields in $\mathcal{E}_1$ and $\mathcal{E}_2$-axes direction.

3.1.3 Simulation and Experimental Results

In this section, we investigate and analyze the solver's performance with different CMs through a variety of experiments using flexible metal rods and polymer-based rods, both with and without magnetic actuation. To generate magnetic fields, the setup as shown in Figure 3.4 consists of two pairs of Helmholtz coils, and each pair consists of two identical electromagnetic coils. They produce two perpendicular fields. In other words, the first pair of coils generates a uniform magnetic field along the $\mathcal{E}_1$-axis. The second pair of smaller coils is placed inside the first pair to produce a field along the $\mathcal{E}_2$-axis. Two cameras are placed next to the setup to monitor the side view of the workspace. For image acquisition, we use both cameras in a stereo vision setup to reconstruct 3D views of the manipulator motion. For the detailed results, refer to Section I.4 in Paper I.

3.1.4 Discussion

Considering the maximum and the mean absolute values of the errors (Table I.8 in Paper I) in each experiment and compared to the manipulators' length, the mean absolute deviations are small, which reflect the model's performance.

It was observed that the frequency of motion depends on the number of nodes considered for the discretization. To make the frequency of motion in simulations compatible with the experiments, a correction factor for frequency shaping was introduced. By employing this correction factor in the simulation, the effect of the number of discretization elements on the frequency of motion

can be eliminated.

Remark 3.1.1. The computational intensity of the proposed dynamic model in this section is intrinsically linked to their implementation approach, aimed at decoding the intricate behaviors of CMs. These dynamic models are comprehensive but computationally heavy, reflecting our prioritization of detail over real-time scalability. As such, they are not immediately poised for real-time control but provide an in-depth understanding that is critical for foundational research. In the pursuit of operability within real-time systems, we anticipate refining these models into reduced-order and data-driven versions. Such versions will balance the fidelity of simulation with the practicality of execution, essential for responsive actions in dynamic settings like surgical robotics. The evolution from our current models to those suitable for real-time control will be explored in the next section, ensuring that critical physical insights are preserved while achieving the efficiencies necessary for live application.

3.2 Real-time Dynamic Model of CMs: RNN-based Prediction

High flexibility and deformity of soft robots, especially CMs, come with a cost of difficulty in steering them in constrained workspaces such as the human body. From a mathematical point of view, the complexity of dealing with CMs is a result of nonlinear PDEs, which are central to expressing the behavior of CMs. Analytical models of soft manipulators help evaluate their motion and determine their workspace in order to be used for control, motion planning, and animation purposes. Proposed approaches in the literature are either kinematics-based models [GDS19; MD11] which are not sufficient for high dynamic motions or finite elements, or differences-based dynamical models [Tar+20; TAR19] which are computationally expensive for real-time applications or fast simulations. Another limitation is that solutions for dynamical models are discrete or not sufficiently differentiable. It is worth noting that having a differentiable solution (i.e., a solution that can be evaluated continuously on the workspace) is crucial in the design process of model-based controllers or observers.

Recently, neural network models are becoming more prevalent among numerical frameworks for approximating PDEs solution [SS18; WHJ17]. Also, these networks were employed to establish physics-informed models in which PDEs are used as to-be-optimized cost functions [RPK17]. In this section, inspired by the time-space integration scheme and using the Lie group variational integration method described in the previous section, the dynamic equations for translation and rotation for each node of a soft CM are decoupled, providing an appropriate structure aimed at developing a real-time modeling algorithm. Afterward, Recurrent Neural Networks (RNNs)-based models are employed to approximate the high-dimensional discretized equations. RNNs with Long Short-Term Memory (LSTM) layers can learn and remember the outputs of nonlinear functions over sequences of inputs. By using internal feedback within LSTM layers, these networks are capable of preserving long-term dependencies.

Essentially, LSTM layers prevent older information from gradually vanishing. LSTM layers consist of LSTM units, which can process sequences of data of any length, for example, poses (positions and orienations). An LSTM unit controls the contributions of each element of its inputs in its output and keeps track of the dependencies between the elements [HS97]. Additionally, external torques and forces (e.g., control inputs, friction, and gravity) are incorporated into the model in a real-time manner for control applications.

The aim here is to propose distributed deep RNNs for capturing and simulating PDEs that describe soft manipulator dynamics. This approach is intended to enable accurate and real-time control compared to existing models. Consider a CM with large deflections described by dynamic equations of motion (as presented in [Tar+20] and [Dem+15]) in the PDEs form as

$$\begin{aligned} \boldsymbol{H}\boldsymbol{\omega}_t + \boldsymbol{\omega} \times \boldsymbol{H}\boldsymbol{\omega} + \boldsymbol{n} \times \boldsymbol{\Lambda}^{-1}\boldsymbol{\phi}_x - \boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x \times \boldsymbol{m} - \boldsymbol{m}_x &= \boldsymbol{\Lambda}^{-1}\boldsymbol{\tau} \\ M\boldsymbol{\phi}_{tt} - \boldsymbol{\Lambda}(\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x \times \boldsymbol{n}) - \boldsymbol{\Lambda}\boldsymbol{n}_x + \boldsymbol{f}^c &= \boldsymbol{f} \end{aligned} \tag{3.1}$$

where $M = \rho \times A$ (ρ and A are the manipulator constant mass density and its cross-section area), $\boldsymbol{\omega} \in \mathbb{R}^3$ is the manipulator's angular velocity, $\boldsymbol{H} \in \mathbb{R}^{3\times3}$ is the manipulator's inertia matrix, $\boldsymbol{\phi} \in \mathbb{R}^3$ is the position of the manipulator's line of centroids in its workspace, $\boldsymbol{\Lambda} \in SO(3)$ denotes the orientation of moving cross-sections at point ϕ. Also, while $\boldsymbol{n} \in \mathbb{R}^3$ and $\boldsymbol{m} \in \mathbb{R}^3$ are the stresses and momenta along the manipulator, $\boldsymbol{f}^c \in \mathbb{R}^3$ represents conservative forces (e.g., gravity). Furthermore, $(\cdot)_x$, $(\cdot)_t$, and $(\cdot)_{tt}$ denote partial derivatives with respect to position, time, and the second partial derivative with respect to time, respectively. Finally, $\boldsymbol{f} \in \mathbb{R}^3$ and $\boldsymbol{\tau} \in \mathbb{R}^3$ are non-conservative forces and torques (e.g., frictions and control inputs) [1].

3.2.1 Proposed RNN-based Modeling and Training Procedure

Using a Lie group variational time integration model, the soft CM dynamics given in Equation (3.1) are discretized where the equations are given in [Tar+20, Sec. 2]. In our study, we discretize the manipulator with equidistant nodes, but this can be changed depending on the application. Figure 3.5 demonstrates a soft CM at time t where x^* is the underformed length of Node n–1. The force $F(x^*, t)$ and torque $\tau(x^*, t)$ are applied to Node n–1 at the position $\phi(x^*, t)$. Also, $\Lambda(x^*, t)$ is the orientation matrix from the frame $\{O\}$ to the frame $\{O_t^{n-1}\}$ attached to the cross-section of Node n–1.

The discrete equations suggest an appropriate structure for the RNNs-based model for a single Node n and the whole manipulator with non-conservative forces and torques as depicted in Figure 3.6. For details about the model, refer to Section II.3 in Paper II.

For the training process, data-sets contain time-sequence inputs and forces and torques applied to each node. Also, for each node, the poses of the node

[1] For the details see [Dem+15].

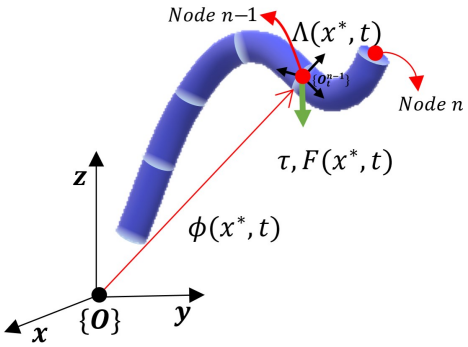


Figure 3.5: A soft manipulator at time t with discretization nodes n and n–1 are shown. $\phi(x^*,t)$ and $\Lambda(x^*,t)$ denote the position and the orientation of cross-section of Node n–1, respectively. In addition, the force $F(x^*,t)$, torque $\tau(x^*,t)$, and the conservative force f^c (e.g., gravity) are applied to Node n–1 at the position $\phi(x^*,t)$.

and its neighbors are considered features, as shown in Figure 3.6. The first and second input layers proceed through LSTM layers and dense layers as hidden layers, respectively. Finally, output layers have resulted from fully connected layers.

3.2.2 Simulation and Experiments

3.2.2.1 Case Study I

Here, we consider different examples and the experimental validation of the proposed RNN-based model. It is worth mentioning that data-sets play a crucial role in the efficiency and accuracy of machine learning-based algorithms. The data acquisition process from a robot in real-world environments is both time- and cost-consuming (implementation of multiple sensors, data filtering, fusion, etc.). As an alternative approach, the required data can be acquired through simulations of high-fidelity models.

The three simulation scenarios are as follows:

- Simulation I: An ellipse without external wrenches, where a cylindrical rod is bent into a circle and its ends are attached to one another. The rod is then deformed into an elliptical shape and released. Due to potential energies in the ellipse, it starts to move without any external disturbances. The goal is to model the behavior of the ellipse resulting from its internal elastic energy.

- Simulation II: A cylindrical rod with external wrenches, where we simulate a rod with a circular cross section that is actuated by external forces such that its tip tracks a square in space. In this example, the goal is to model

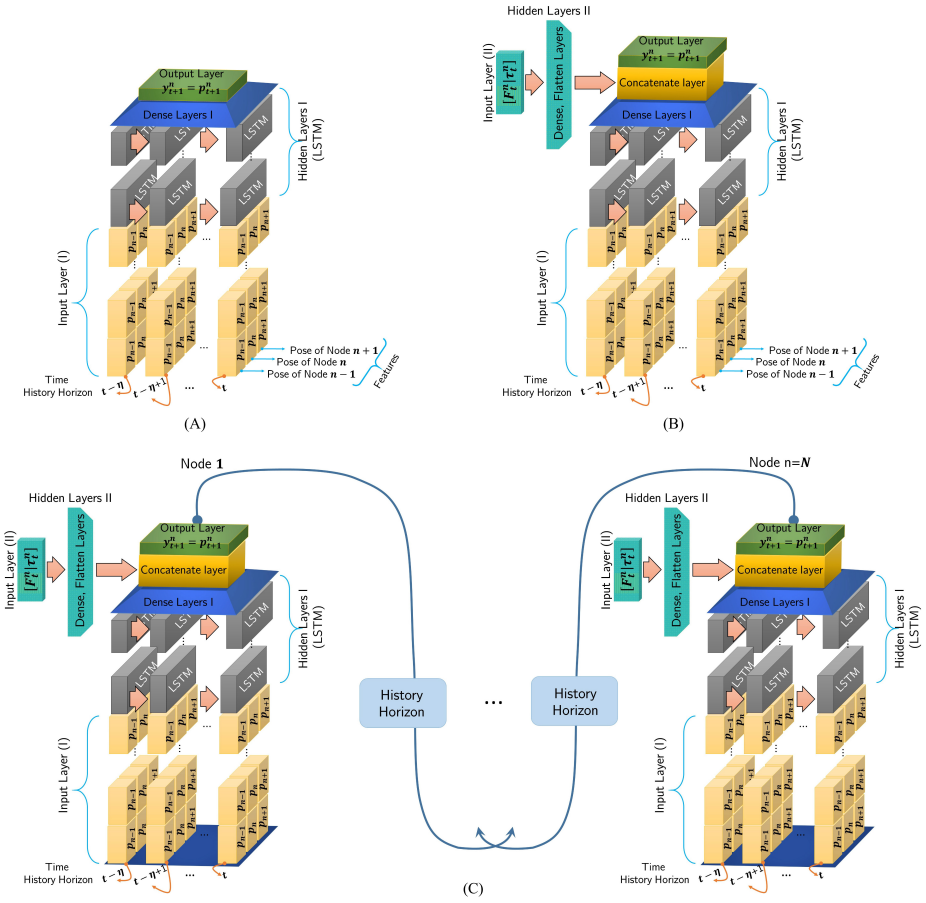


Figure 3.6: Recurrent Neural Network-based model, length of the time history horizon is determined by η and features composed of adjacent nodes pose: **(A)**: Poses of Nodes n–1, n, and n+1 are the input layer and no forces or torques are applied to the node. **(B)**: The first input layer is composed of poses $\boldsymbol{p_{n-1}}$, $\boldsymbol{p_n}$, and $\boldsymbol{p_{n+1}}$ at time history horizon $[t\text{–}\eta, t\text{–}\eta\text{+}1, \cdots, t]$ and the second input layer includes forces and torques $[\boldsymbol{F_t^n}, \boldsymbol{\tau_t^n}]^T$ which are incorporated into the model through the Hidden Layers II (dense and flatten layers). **(C)**: Proposed RNN-based models of the CM with N nodes including Input, Hidden, and Output Layers. A history of each node output is used as an input for adjacent nodes. Nodes poses (Input Layer I) and forces and torques (Input Layer II) through the Hidden layers I and II are proceed and concatenated together.

the behavior of the rod, which results from applied external forces on its end-effector.

- Simulation III: A cylindrical rod with and without external wrenches, where we formed a semi-circular shape with a cylindrical rod. A force is applied to the middle node for 0.5 s, and then the force is removed. The idea is to model the behavior of the rod resulting from applied external forces and internal elastic energy.

where we includes detailed comparisons regarding computational requirements between the RNN-based approach and the classical Lie group variational integrator. These comparisons, emphasizing the focus on position prediction, demonstrate the RNN model's superior efficiency and reduced computational demands, suitable for real-time applications.

For the experiment, we fabricated a soft manipulator on which magnetic fields are used to produce the necessary forces and torques. Furthermore, to show the performance of the algorithm, results from the presented method and a Cosserat rod-based theoretical model —presented in Appendix II.A in Paper II— are compared to show the efficiency of the proposed RNN-based model.

Remark 3.2.1. The focus is on demonstrating a real-time dynamic predictive model for soft CMs that leverages neural network-based strategies optimized for computational efficiency in real-time applications. The choice to compare our model's performance with the Cosserat rod model was driven by the Cosserat rod theory's relevance and established use in modeling the mechanics of soft CMs, particularly for its quick predictive capabilities which are crucial in dynamic and real-time scenarios. By comparing our proposed model with an established and well-used fast model in the literature, we aim to provide evidence that our approach can outperform even these fast models, underscoring the efficacy and efficiency of our neural network-based solution.

Our model's architecture is specifically tailored to provide rapid predictions essential for real-time control, distinguishing it from the objectives of Lie group variational integrators, which, while powerful for preserving geometric properties, are typically not optimized for speed. Thus, the comparison with the Cosserat rod theory was more pertinent to the scope and aims of our work, highlighting the advantages of our model in terms of computation time and real-time application suitability.

Moreover, we have conducted three comparative simulations to showcase the real-time capabilities of our proposed method against Lie group variational integrators. These simulations illustrate that our neural network-based approach maintains a competitive edge in speed and performance.

A soft CM is fabricated from urethane rubber, Polymer Matrix Composite 770 (PMC-770, Smooth-On Inc., USA) and neodymium (NdFeB) block magnets, and the experimental setup consists of 6 stationary electromagnets surrounding a spherical workspace of 100 mm diameter. Figure 3.7 shows the soft manipulator, its segmented shape reconstruction, and the setup of the experiment.

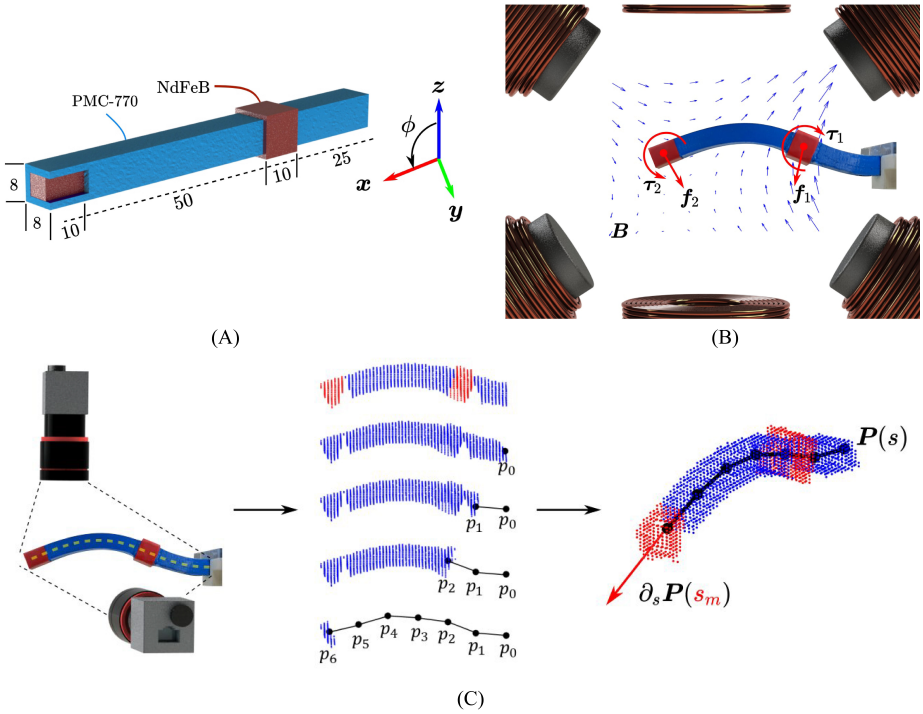


Figure 3.7: **(A)**: Polymer matrix composite 770 (PMC-770) beam continuum manipulator with embedded neodymium (NdFeb) magnets located at tip and intermediate positions. Dimensions are given in millimeter. **(B)**: Experimental setup consists of 6 stationary electromagnets and contains a segmented photograph of the final shape manipulator. The flexible PMC-770 and rigid NdFeb sections of the manipulator are blue and red, respectively. Six electromagnets generate a magnetic field ($\boldsymbol{B}$) in the workspace, exerting torques and forces ($\boldsymbol{\tau}_m, \boldsymbol{f}_m,\ m = 1, 2$) on the magnets, which deforms the CM to its final shape at the time $t = 340$ s. **C**: Representation of the shape reconstruction algorithm used for shape feedback. The manipulator is recorded with a stereo vision setup. The manipulator body is represented by a 3D spatial point cloud. The manipulator centerline, characterized by parameter $s \in [0, L]$, is approximated by $N + 1$ points ($\{\boldsymbol{p}_0, \ldots, \boldsymbol{p}_N\}$). A 3D polynomial fit ($\boldsymbol{P}(s)$) is made through the points, and the magnet orientation at an assumed constant centerline position s_m derived from the local gradient of the polynomial fit.

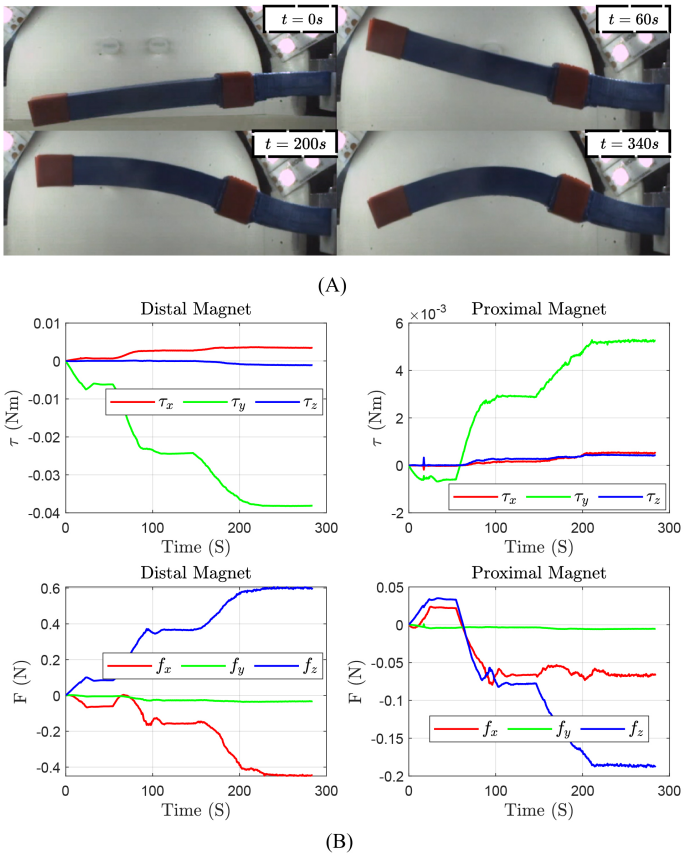


Figure 3.8: **(A)**: Initial and time-evolved configurations. **(B)**: Applied torques and forces on the distal and proximal magnets.

When the manipulator is subjected to an external magnetic field, the embedded magnets experience forces and torques. This causes the flexible portions of the manipulator composed of the PMC to undergo elastic deformation. The applied magnetic forces and torques, together with the initial and a few time-evolved configurations, are shown in Figure 3.8.

For rods properties, simulation parameters, the structure of the proposed models and data-sets, details of the training processes, and results on the prediction and real-time performances in the simulations and experiments, refer to Section II.4 and Section II.5 in Paper II.

To sum up, the experiment demonstrated that not only can the presented RNN-based model outperform classical modeling approaches such as the Cosserat rod model, but it also shows possibilities for using the model in practice for closed-loop control applications.

Remark 3.2.2. Table 3.1 summarizes the training and validation time for the

simulations and experiment.

Simulation examples	**Training and Validation Time (S)**
Simulation I	33
Simulation II	322
Simulation III	11
Experiment	**Training and Validation Time (S)**
Experiment	1.1

Table 3.1: Training and validation times for the simulations and experiment are given in seconds. The training and validation of models were conducted on a 16 GB, 1.99 GHz Intel i7 machine running Windows 10.

To sum up the findings up to this point, It is worth mentioning that increasing the size of history horizons in the training stages may reduce the error to some extent, but on the other hand, it makes the model slower. Based on conventional dynamical models, the length of the history size should be at least 2.

To reach state-of-the-art performance, i.e., having less error and a faster model simultaneously, one may prefer varying batch sizes in the training and run-time phases. As a suggestion, we can use different batch sizes for training and run-time stages. A model can be trained with appropriate batch sizes such that the model's performance meets the given criteria. Afterward, one can create a new network with the pre-trained weights compiled with a batch size of 1.

The performance, i.e., the convergence and stability, of the presented algorithm, unlike conventional algorithms, is independent of the number of nodes considered for the whole manipulator. To be specific, in the analytical model, there might be a need for several discretization nodes to achieve a convergent solution with a specific tolerable error; however, in the RNN-based model, only specific points or points of interest (e.g., two actuation points in the experiment) are considered. In other words, in the experiment, 13 nodes (4 for each flexible subsection, 2 for each magnet, and 1 for the base) were chosen for solving the Cosserat rod model, but only two nodes were selected for the RNN-based model. However, the complexity of dynamical systems (i.e., PDEs) affects the complexity of the architecture used in the RNN-based model, i.e., the number of layers and LSTM units, and generally how deep the model is. Nevertheless, the suggested model suits parallel implementation and can benefit from a high bandwidth for closed-loop control applications. Furthermore, the architecture of the proposed RNN-based model can be optimized by reducing the number of layers and trainable parameters to maximize the achievable bandwidths.

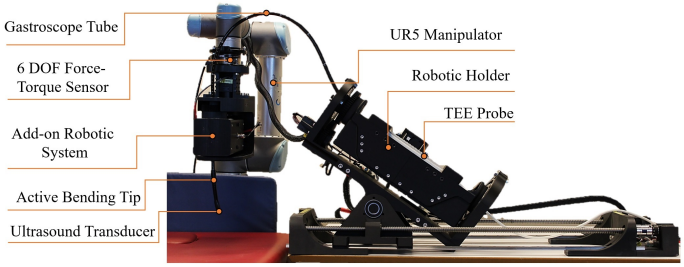


Figure 3.9: The constructed robotic-assisted Transesophageal Echocardiography System. Different components of the system are as follows: UR5 Manipulator, Robotic Holder, TEE Probe, Gastroscope Tube, 6 DoFs Force-Torque Sensor, Add-on Robotic System, Active Bending Tip, and Ultrasound Transducer.

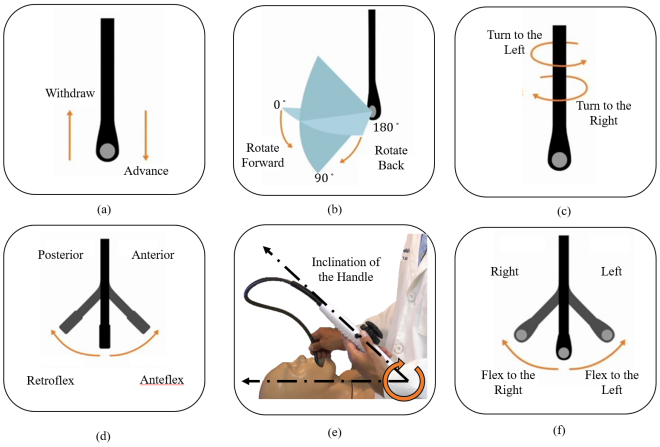


Figure 3.10: The terminology used to describe the manual manipulation of the TEE probe.

3.2.2.2 Case Study II

The same line of idea discussed in Section 3.2 is used for the modeling of a flexible cable-driven Transesophageal echocardiography (TEE) probe. The proposed robotic-assisted transesophageal echocardiography system is given in Figure 3.9. The objective is to predict the positions and orientations of the TEE transducer based on motor inputs within the workspace. The terminology used to describe the manual manipulation of the TEE probe is given in Figure 3.10 and demonstrated on the proposed system in Figure 3.11.

In order to generate a data-set to train the algorithm on, several considerations should be taken into account:

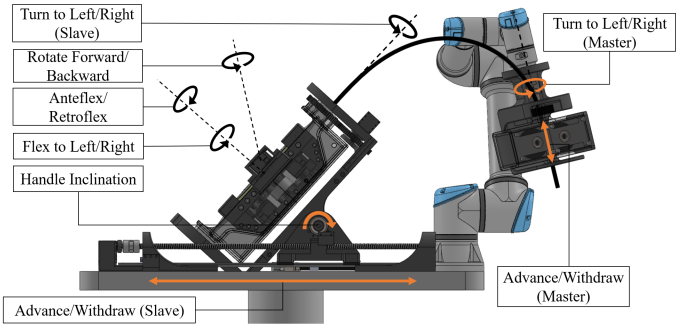


Figure 3.11: The conceptual design of the proposed robotic system where different movement of TEE is depicted.

- During the insertion of the TEE transducer, the gastroscope tube has to bend almost 90 degrees from the mouth to the esophagus. This bending causes varying levels of friction between the cables and the inner surface of the gastroscope tube, depending on the bend angle. The data-set aims to capture this effect by considering bends of the gastroscope tube from 0 to 90 degrees. This will account for the impact of different gastroscope shapes on the creation of dead zones. The data-set primarily documents the kinematic behavior of the probe, and includes three distinct tip bend angles for the TEE probe: 0, 45, and 90 degrees.

- The active bending portion of the TEE probe loses rigidity and becomes limp when its wires are in their initial, relaxed state. This leads to issues of hysteresis and back relaxation, making the control of the TEE transducer difficult when it's in its initial position. Consequently, the data gathering process should incorporate considerations for these hysteresis and back relaxation phenomena in the active bending part of the probe.

- Based on previous research [Saj+22], the dead zone of the small wheel (which controls flex to Left/Right as shown in Figure 3.11) is influenced by the angle of the large wheel (which is in charge of manipulating Anteflex/Retroflex as shown in Figure 3.11), and the reverse is also true. When both wheels are in operation, the accuracy of the geometric kinematic model decreases in certain trajectory segments where the mechanism resides within the dead zones of both wheels. Therefore, the data-set needs to encompass the probe's movement when both the small and large wheels are activated.

- The data-set should include a variety of ranges of motion along the $x-$, $y-$, and $z-$axes, as well as axis rotations.

The square-shaped trajectory in different rotations (Turn to Left/Right) and translations (Advance/Withdraw) has been chosen for building the data-set to

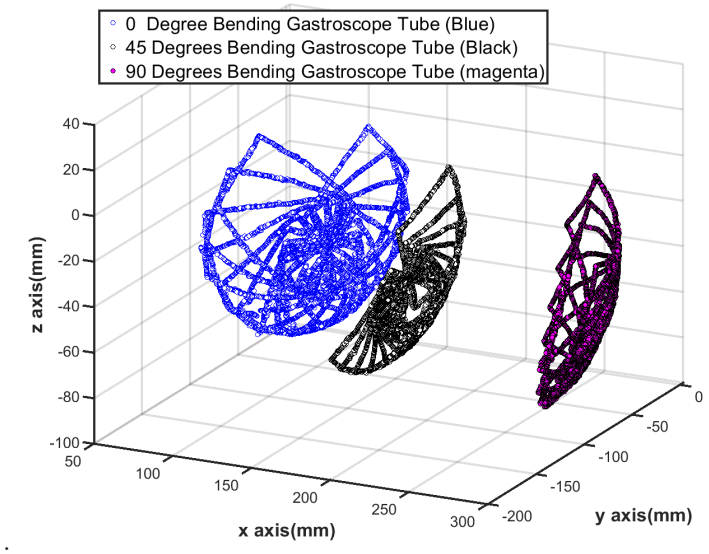


Figure 3.12: TEE tip positions in different data-sets.

incorporate the essential considerations, as illustrated in Figure 3.12. The focus was on the change of the gastroscope tube inside the esophagus, and 20010 samples were recorded at zero degree bending, 14674 samples for 45 degree bending, and 17342 samples for 90 degree bending to include different dead zones caused by the different shapes of the gastroscope. Furthermore, for each square trajectory, the probe is manipulated toward the initial position to incorporate the hysteresis and back relaxation of the active bending part in the data-set, resulting in different initial conditions for each trajectory.

The pose of the TEE transducer node is augmented in a 1-by-7 vector at each time step. The size of the historical horizons has been determined to be two $\eta = 2$. As a result, 2-by-7 tensors are produced for each time step and fed to the model as an input layer. The motor position samples that are applied as another input layer as 2-by-4 tensors go through the same preparation process. According to Figure 3.6, the input layers are composed of tensors of size (Batch Size $\times$ 2 $\times$ 7) and (Batch Size $\times$ 2 $\times$ 4), respectively. All layers' initial dimension is reserved for batch sizes for training purposes, and batch size 1 was chosen. The results of estimation for scenarios 0, 45, and 90 degree bends are depicted in Figure IV.17, Figure IV.18, and Figure IV.19 and explained in detail in Section IV.4.3.

3.3 A Rapid Physics-base Kinematic Model

Modeling of soft robots is computationally heavy due to complex geometries and coupled actuation mechanisms such as thermal, electrical, or magnetic actuators. In other words, soft robots may be coupled with other physical

fields for actuation purposes, and therefore these physical fields should be taken into account in the modeling. A rapid multiphysics simulation framework is proposed for magnetically actuated CMs within an environment similar to the Gastrointestinal (GI) tract, which can be used in closed-loop control applications to reduce the reality gap.

Firstly, a model for magnetic wrenches is developed where each permanent magnet is modeled as one or multiple dipoles depending on its size. Then, for the guidewire, several joints are considered on the wire and modeled as rotational springs (to capture the bending potential energy), and segments in between the joints are modeled as a linear spring to capture the stretch of the wire. Finally, the guidewire into which a permanent magnet is embedded as an actuation point will be shaped to go through a GI tract structure by applying external magnetic fields through single or multiple permanent magnets. The formation of the wire will be discussed in the next chapter.

3.3.1 Modeling of Magnetic Wrenches

The aim here is to develop a light model for calculating the interacting forces and torques between two dipoles — for simplicity, magnets are represented by dipoles —. Forces at position $\bar{d}$ resulting from the interaction of two dipoles with magnetic moments $\bar{\mu}_1$ and $\bar{\mu}_2$ are the same in magnitude with opposite directions can be written as $F = \frac{3\mu_0}{4\pi d^5}\left[\bar{n}_1\bar{\mu}_2 + \bar{n}_2\bar{\mu}_1 + <\bar{\mu}_1, \bar{\mu}_2> \bar{d} - \frac{5}{d^2}\bar{n}_1\bar{n}_2\bar{d}\right]$ where $\bar{n}_1 =< \bar{\mu}_1, \bar{d} >$, $\bar{n}_2 =< \bar{\mu}_2, \bar{d} >$, $d = \|\bar{d}\|_2$, $\|\cdot\|_2$ denotes norm 2, and $<\cdot>$ is used to denote the dot product or scalar product between two vectors. The magnetic torque of the dipole $\bar{\mu}_2$ acting on the dipole $\bar{\mu}_1$ is defined as $\bar{\tau}_1 = \bar{\mu}_1 \times \bar{B}_2$ in which $\bar{B}_2$ is the magnetic field of the dipole $\bar{\mu}_2$ and is defines as $\bar{B}_2(\bar{d}) = \frac{\mu_0}{4\pi}\left[\frac{3\bar{d}}{d^5}\bar{n}_2 - \frac{\bar{\mu}_2}{d^3}\right]$. In addition, the magnetic torque of the dipole $\bar{\mu}_1$ acting on the dipole $\bar{\mu}_2$ can be found in a similar way.

The effects of a magnetic force resulting from an attraction field between two aligned dipoles and the magnetic torque resulting from the misalignment of two dipoles are shown in Figure 3.13 and Figure 3.14, respectively. In other words, by moving the free dipole toward the constrained dipole, the beam bends due to the magnetic force resulting from an attraction field (Figure 3.13). Furthermore, by rotating the free dipole in place, the beam deflects from its original position due to the magnetic torque resulting from the misalignment of two dipoles (Figure 3.14).

An external magnet is modeled as a set of 2000 dipoles rotating around the y-axis and Figure 3.15 shows the resulting flux and the applied force on the tip dipole, and the projected force on the spline.

3.3.2 Guidewire Modeling

A guidewire is modeled as multiple segments and joints, which are modeled as extensible and rotational springs, respectively. The embedded magnet in the tip of the guidewire is considered a rigid body. Since guidewires are usually thin,

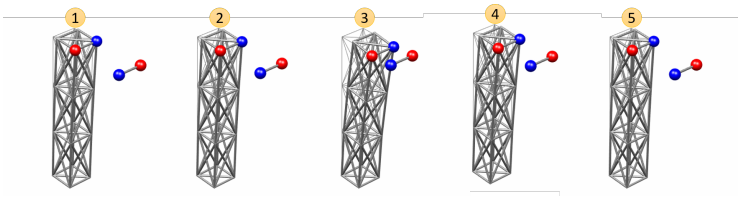


Figure 3.13: By moving the free dipole toward the constrained dipole, the beam bends due to the magnetic force resulting from an attraction field.

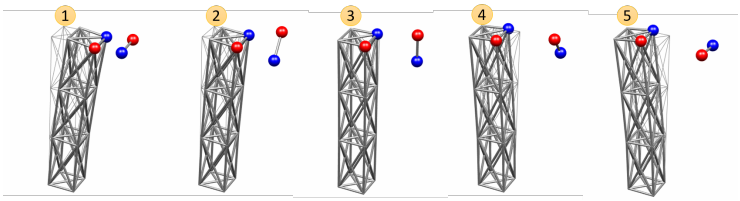


Figure 3.14: By rotating the free dipole in place, the beam deflects from its original position due to the magnetic torque resulting from the misalignment of two dipoles.

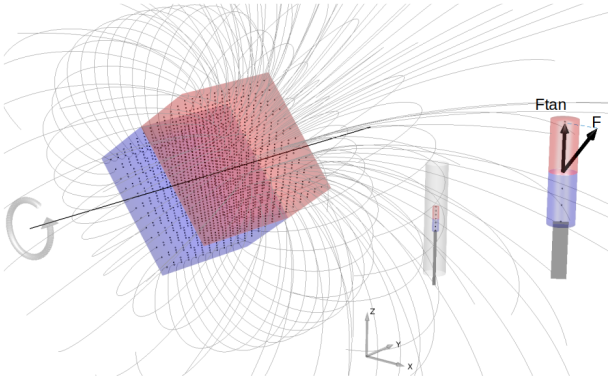


Figure 3.15: A rotating external magnet modeled as 2000 dipoles. The resulting flux and the applied force on the tip dipole and the projected force on the guidewire's centerline are shown.

twisting is not considered in the modeling procedure. The model is shown in Figure 3.16, and the modeling procedure is summarized in Algorithm 1.

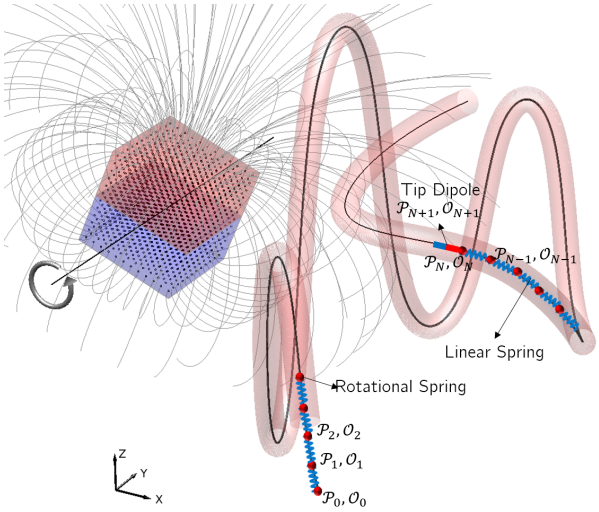


Figure 3.16: A guidewire is partitioned into multiple segments and joints: Each segment is modeled as a linear spring and joints are modeled as rotational springs.

The steering setup, the formation of a guidewire, and the presented optimal control will be discussed in the next chapter.

3.4 Findings and Implications

In this chapter, we have proposed and validated dynamic modeling approaches for CMs. High-fidelity models are helpful for explaining and predicting the behavior of a system with complex dynamics. However, due to computational constraints, these models may not be employed for closed-loop control purposes in a real-time implementation of robotic applications. Additionally, recent developments in computer simulation demand superior, robust, and efficient numerical frameworks compared to traditional approaches. Discrete geometric mechanics, which are employed in this chapter, provide a systematic method to cope with the complexity of CMs' dynamics. The necessity of guaranteeing robots' performance in sensitive applications such as minimally invasive surgeries requires the use of pre-existing knowledge or a model in control architecture to obtain guaranteed and reliable behavior in the presence of disturbances and uncertainties. Although model-free control approaches are easy to implement, they do not provide or ensure any performance level or high control-loop bandwidths.

In Section 3.1, the estimation and model validation problems of CMs' dynamics using Lie group variational integrators were studied. Using magnetic actuation, dynamic and static experiments were conducted on manipulators with

Initialize:
Define a spline curve within the GI tract structure;
Orientation of joints in rest state on spline: $\{\mathcal{O}_0, \cdots, \mathcal{O}_{N+1}\}$;
Position of joints in rest state on spline: $\{\mathcal{P}_0, \cdots, \mathcal{P}_{N+1}\}$;
Initial velocity of joints is zero;
Boundary conditions are considered free at both ends;
while *simulating* **do**

i. Apply external magnetic forces on tip dipole (attached magnet to the tip) alongside with the spline;

ii. Integrate tip magnet position;

iii. Enforce tip magnet position coupling to the rest of the guidewire;

iv. Calculate spring forces between guidewire segments due to segment length deviations by using a spring constant;

v. Calculate bending forces between wire segments (at joints);

vi. Integrate joints poses;

vii. Project all joints forces to the direction of the corresponding guidewire segment (alongside with the spline);

viii. Update joints position;

ix. Snap all joints onto the closest spline;

end

Algorithm 1: Guidewire simulation.

rigid and soft materials (e.g., aluminum and PDMS) to illustrate the validity of the presented algorithm for a wide range of experiments.

Due to the lack of knowledge about friction and damping, distributed predictive filters were designed to provide information about the unknown signals. Therefore, the dynamical model equipped with the estimation algorithm is a self-contained generic model for CM integration, which provides us with a systematic approach to employing optimal control theory for realistic trajectory planning in the presence of user- or environment-specified constraints. The design of a controller and the parallel variational integration algorithm are to be investigated in future work.

In Section 3.2, a distributed architecture for modeling complex dynamical systems by using multiple light-weight RNN-based models is proposed. As a result, the architecture would be easier to design and debug and would also benefit from faster convergence compared to one large network. Furthermore, large networks may take longer to train, they may not show acceptable performance, and readjusting (hyper-)parameters and restarting the training process might be necessary.

The evaluations showed that incorporating poses of adjacent nodes and also wrenches as a separate input might help to have, to some extent, a generalizable model rather than just purely learning the structure of data. However, supervised learning methods likely tend to preserve the structure of data, and these models might not entirely respect the underlying physics (conservation laws). In other words, these methods might not be wholly physics-aware and applicable for untrained or unprecedented dynamics or geometries without any adjustment, re-training, or using techniques such as transfer learning, etc. One possible and interesting solution [LLF98; PU92; RPK19] to overcome this problem and move toward fully physics-aware neural networks is revisiting lost functions for the training process. To be specific, the idea is finding solutions for PDEs given in Equation (3.1), i.e., $\Lambda(x,t)$ and $\phi(x,t)$ for sufficiently large number (e.g., N_f) of pairs $(x_i,t_i) \in (0,L)\times[0,T]$ in which L is the unreformed length of the manipulator and parameter T is a user-defined time. Considering Equation (3.1), a neural network can learn by minimizing the mean squared error loss.

$$\frac{1}{N_f}\left(\sum_{i=0}^{N_f}\left\|\boldsymbol{J}\boldsymbol{\omega}_{t_i}+\boldsymbol{\omega}\times\boldsymbol{J}\boldsymbol{\omega}+\boldsymbol{n}\times\boldsymbol{\Lambda}^{-1}\boldsymbol{\phi}_{x_i}-\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_{x_i}\times\boldsymbol{m}-\boldsymbol{m}_{x_i}-\boldsymbol{\Lambda}^{-1}\boldsymbol{\tau}\right\|^2_{\{x_i,t_i\}}\right.$$
$$\left.+\sum_{i=0}^{N_f}\left\|M\boldsymbol{\phi}_{t_it_i}-\boldsymbol{\Lambda}(\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_{x_i}\times\boldsymbol{n})-\boldsymbol{\Lambda}\boldsymbol{n}_{x_i}+\boldsymbol{f}^{nc}-\boldsymbol{f}\right\|^2_{\{x_i,t_i\}}\right)$$

This modified loss function enforces the structure imposed by Equation (3.1) for a large number (e.g., N_f) of pairs $(x_i,t_i)\in(0,L)\times[0,T]$, and the trained neural network will be aware of the governing PDEs.

In conclusion, this section described an approach for the real-time prediction of dynamics for general continuum soft manipulators based on machine learning techniques and Lie group variational integration methods. Poses of a soft,

polymer-based manipulator, in the presence of conservative and non-conservative wrenches, are predicted and validated experimentally. The comparison results of the proposed model and a well-known model for CMs, i.e., Cosserat rod theory, are also provided, revealing the practical effectiveness of the proposed model. The presented method can be extended to different soft robots with different shapes and materials. In addition, the training of physics-aware neural networks for solving PDEs and the procedure of a model-based controller design are topics of research to be studied in future work.

References

[BS99] Bobenko, A. I. and Suris, Y. B. "Discrete Time Lagrangian Mechanics on Lie Groups with an Application to the Lagrange Top". In: *Communications in mathematical physics* vol. 204, no. 1 (1999), pp. 147–188.

[Dem+15] Demoures, F. et al. "Discrete variational Lie group formulation of geometrically exact beam dynamics". en. In: *Numerische Mathematik* vol. 130, no. 1 (May 2015), pp. 73–123.

[GDS19] Grazioso, S., Di Gironimo, G., and Siciliano, B. "A geometrically exact model for soft continuum robots: The finite element deformation space formulation". In: *Soft robotics* vol. 6, no. 6 (2019), pp. 790–811.

[Hai+06] Hairer, E. et al. "Geometric numerical integration". In: *Oberwolfach Reports* vol. 3, no. 1 (2006), pp. 805–882.

[HS97] Hochreiter, S. and Schmidhuber, J. "Long short-term memory". In: *Neural computation* vol. 9, no. 8 (1997), pp. 1735–1780.

[LLF98] Lagaris, I. E., Likas, A., and Fotiadis, D. I. "Artificial neural networks for solving ordinary and partial differential equations". In: *IEEE transactions on neural networks* vol. 9, no. 5 (1998), pp. 987–1000.

[LOL14] Leitz, T., Ober-Blöbaum, S., and Leyendecker, S. "Variational Lie Group Formulation of Geometrically Exact Beam Dynamics: Synchronous and Asynchronous Integration". In: *Multibody Dynamics*. Springer, 2014, pp. 175–203.

[Lu95] Lu, P. "Optimal predictive control of continuous nonlinear systems". In: *International Journal of Control* vol. 62, no. 3 (1995), pp. 633–649.

[MD11] Mahvash, M. and Dupont, P. E. "Stiffness control of surgical continuum manipulators". In: *IEEE Transactions on Robotics* vol. 27, no. 2 (2011), pp. 334–345.

[MR13] Marsden, J. E. and Ratiu, T. S. *Introduction to mechanics and symmetry: a basic exposition of classical mechanical systems*. Vol. 17. Springer Science & Business Media, 2013.

[PU92] Psichogios, D. C. and Ungar, L. H. “A hybrid neural network-first principles approach to process modeling”. In: *AIChE Journal* vol. 38, no. 10 (1992), pp. 1499–1511.

[RPK17] Raissi, M., Perdikaris, P., and Karniadakis, G. E. “Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations”. In: *arXiv preprint arXiv:1711.10561* (2017).

[RPK19] Raissi, M., Perdikaris, P., and Karniadakis, G. E. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational Physics* vol. 378 (2019), pp. 686–707.

[Saj+22] Sajadi, S. M. et al. “Design, kinematic modeling, and validation of a robotic-assisted transesophageal echocardiography system”. In: *2022 IEEE/SICE International Symposium on System Integration (SII)*. IEEE. 2022, pp. 250–257.

[Sim85] Simo, J. C. “A finite strain beam formulation. The three-dimensional dynamic problem. Part I”. In: *Computer methods in applied mechanics and engineering* vol. 49, no. 1 (1985), pp. 55–70.

[SS18] Sirignano, J. and Spiliopoulos, K. “DGM: A deep learning algorithm for solving partial differential equations”. In: *Journal of computational physics* vol. 375 (2018), pp. 1339–1364.

[Tar+20] Tariverdi, A. et al. “Dynamic modeling of soft continuum manipulators using lie group variational integration”. In: *Plos one* vol. 15, no. 7 (2020), e0236121.

[TAR19] Till, J., Aloi, V., and Rucker, C. “Real-time dynamics of soft and continuum robots based on Cosserat rod models”. In: *The International Journal of Robotics Research* vol. 38, no. 6 (2019), pp. 723–746.

[WHJ17] Weinan, E., Han, J., and Jentzen, A. “Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations”. In: *Communications in Mathematics and Statistics* vol. 5, no. 4 (2017), pp. 349–380.

Chapter 4

Control Strategies for CMs

This chapter deals with the research question **RQ.2** described in Section 1.4 and is adapted from Paper III and Paper V. Figure 4.1 gives an overview of Chapter 4.

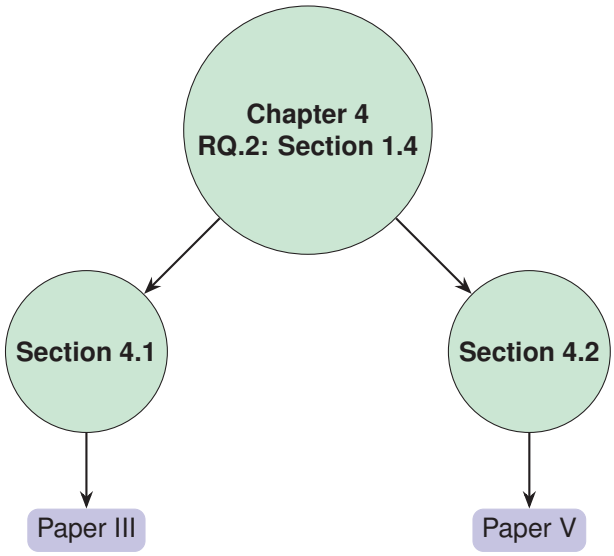


Figure 4.1: An overview of Chapter 4.

Soft, flexible materials and rigid, permanent milliscale magnets or coils are typically used to fabricate magnetically-actuated CMs. When these CMs are exposed to an external magnetic field, embedded magnets or coils experience forces and torques (wrenches), which cause flexible portions of the CMs to deform. Automated steering of these manipulators can be seen as a problem where embedded magnets need to be ultimately controlled. However, once the tip of a CM is guided onto a specific trajectory, the rest of the CM's body tends to follow that trajectory closely. Nevertheless, actuation, tracking, and closed-loop positioning of these embedded rigid bodies, and thus, CMs, remain a challenge, yet crucial in ensuring adequate reachability and precision. Miniature robotics advancements have resulted in small-scale robots capable of targeted interaction within the human body. Nonetheless, in a clinical setting, the use of this technology is still constrained by the auxiliary infrastructure necessary for the actuation of small robots. Using a combination of Reinforcement Learning (RL) algorithms and a classical switching control approach —presented in Paper III—, we show that small-scale particles can be steered within a constrained environment

in the presence of disturbances by providing adequate external magnetic forces. In addition, two other classical path-finding approaches, Attractor Dynamics and Execution Extended Rapidly-Exploring Random Trees (ERRT), are also investigated and compared to the RL-based method. Finally, a rapid kinematic model developed in Section 3.3 and Paper V is used to steer a guidewire through a Gastrointestinal (GI) tract structure by applying an external magnetic field resulting from a rotating permanent magnet around only one axis.

4.1 RL-based Switching Controller for an Untethered Magnetic Particle

Due to the potential of millirobotics to interact with the human body at millimeter sizes, the utilization of these tiny robots as innovative surgical instruments has gotten a lot of interest alongside the research on CMs. In recent years, there have been substantial efforts to develop soft, untethered robots capable of performing targeted drug delivery [Lu+18; VS16] directly to a tumor or other disease sites precisely and non-invasively as traditional approaches such as pills and intravenous therapy have been the primary method of drug delivery for decades [Tiw+12; WJL15].

Unlike CMs that can traverse large distances within workspaces inside the human body due to their continuously bending and compliance structure, positioning and motion control for untethered milliscale robots over large distances is difficult. One of the difficulties for small-scale robots moving toward diseased tissues is disturbances introduced by bloodstream or other biophysical barriers and obstacles [Bou12; Sch+19].

It should be noted that the miniaturization of conventional robots is limited mostly by the mechanical structures required to allow them to navigate. By contrast, magnetic actuation is a mechanism that can remotely and wirelessly actuate robots without necessitating specialized structures directly in the robots, thus substantially reducing their size and complexity. Reducing the size of robots enables them to move to difficult-to-reach areas inside biological bodies by applying an external magnetic field that harmlessly penetrates patients' bodies and applies wrenches on the robots.

Magnetic manipulation of micro/milliscale robots with an application such as magnetic drug delivery has been an increasingly popular approach where drug particles in the body are carried and manipulated by external magnetic fields to reach a targeted location [Ami+17; Hos+17a; Hos+17b; Hos+18]. However, investigating existing results on magnetic control of small-scale robots reveals that these studies employ either open-loop or traditional closed-loop control mechanisms that may show degraded performance or fail in uncertain environments. In other words, in practice, the methods are not robust in the presence of disturbances. In addition, the existing studies do not take into account any obstacle avoidance strategies in the design of controllers or trajectory planners, and then these methods may not be robust enough for realistic scenarios.

Based on the issues mentioned above, more practical approaches to magnetic control of small-scale robots require investigation. Although designing online path planners together with appropriate, robust controllers is a problem-specific approach, deep Reinforcement Learning (RL) methods can handle both the trajectory planning and control tasks simultaneously and can be generalized to be used in various workspaces. We propose to employ a deep RL algorithm for magnetic position control of an object in a constrained environment.

Deep reinforcement learning methods have been widely applied in computer games and simulations. However, employing these algorithms for practical, real-world applications such as robotics becomes challenging due to the difficulty in obtaining training samples. The effort here predominantly focuses on bridging the gap between simulations and the real-world implementation of a reinforcement learning algorithm for a robotic application in the context of miniaturized drug delivery robots and robotic capsule endoscopes. In addition, the derivation and experimental validation of a reinforcement learning-based algorithm for controlling a magnetically-actuated small-scale robot within a simplified model of the large intestine in the presence of disturbances are presented. We demonstrate the possibility of training a high fidelity reinforcement learning algorithm fully within a simulated environment before deploying it as-is in a real-world scenario through carrying out different experiments. Implementing the presented control framework complements a large body of this chapter, and the results offer a feasibility study of using a reinforcement learning algorithm in practice.

To evaluate results of the RL-based mechanism, two classical approaches, Attractor Dynamics[KB12] and Execution Extended Rapidly-Exploring Random Trees (ERRT)[BV02] are tested through simulations and experiments, in which the results confirm that the RL-based method achieves comparable performance while being a more robust and generic solution to deploy in dynamic, complex environments.

4.1.1 Preliminaries on RL

A technique to tackle control and tracking problems known as RL is based on learning control policies via repeated trial-and-error interactions between a robot and its environment. RL is a process in which a robot's behavior is taught to maximize its environment's predicted sum of reward via feedback signals.

In the classic Q-Learning approach [WD92], learning Q-values is the primary objective. If the robot starts from a specific state-action combination and follows a particular policy, Q-values are the predicted cumulative rewards that the robot will get over time. A significant milestone was achieved by introducing the Deep Q-Networks (DQN) algorithm [Mni+13]. Unlike its table counterpart, DQN represents all of the Q-values as a multi-layered neural network, rather than storing them all in a look-up table as is done with the table variant.

The deep reinforcement learning community has made gradually several improvements to the DQN algorithm [Mni+13] to cope with its shortages, for example, Double Deep Q-Learning Networks (DDQN) [VGS15], Prioritized DQN [Sch+15], Dueling DQN [Wan+16], Distributional DQN [BDM17], Noisy

DQN [For+17]. Finally, Rainbow [Hes+17] incorporates the advances made in earlier versions.

4.1.2 Method: Reinforcement Learning, Attractor Dynamics, and ERRT-based Switching Controller

This section explains the proposed control architecture, consisting of two controllers (Sub-controllers 1 and 2) and a switching rule. Sub-controller 1 consists of the inverse kinematics of the robot and keeps the magnetic particle within the immediate vicinity of the robotic arm end-effector. Once the particle is far enough from the end-effector, Sub-controller 1 takes the robotic arm to the particle position, which is fed back to Sub-controller 1 from an optical tracking system. As soon as the particle is again close enough to the end-effector, an appropriate switching law makes Sub-controller 2 to control the robot arm to carry the particle toward an objective inside the environment using a magnetic field produced by an attached permanent magnet on the end-effector. In Sub-controller 2, three approaches, i.e., reinforcement learning, attractor dynamics, and ERRT algorithms, are used to carry the particle through a disturbed environment toward a given target position.

4.1.2.1 Reinforcement Learning-based Approach

Sub-controllers 1 and 2 and the switching law are shown in Figure 4.2. The control system should be resilient to losing the particle (when it ceases to react to the field) and capable of automatically retrieving it if this happens. Sub-controller 1 is in charge of maintaining the magnetic particle's proximity to the end-effector using information obtained from an optical tracking system. The control-loop mechanism is shown in Figure 4.3a when Sub-controller 1 is enabled.

Sub-controller 2 drives the robotic arm to a preferred location inside a confined (intestine-like) environment. Notably, this controller does not include inputs from the optical tracking system, and the robot sees the surroundings partially through the RGB camera on its end-effector. The control-loop mechanism is illustrated in Figure 4.3b when Sub-controller 2 is triggered. Inspired by [Hes+17], for Sub-controller 2, a modified Rainbow algorithm with Quantile Huber loss derived from the Implicit Quantile Networks (IQN) algorithm [Dab+18] together with ResNet [He+16] is employed. Since Mnih's publication of DQN [Mni+15], the Huber loss has been utilized for training the family of DQN RL algorithms. Obando-Ceron's comparison research in [OC20] shows that the Huber loss outperforms the more conventional mean-squared error loss.

Observations fed to Sub-controller 2 are two-channel downsampled images with a dimension of 84×84. A set of 15 by 15 pixels in the center of each image is considered, and then the mean of the pixel values in that square is calculated. An obstacle is detected when the pixels' average value is less than 150. Four discrete actions are considered to control the robotic arm end-effector in the xy-plane in two dimensions by setting a new position for the end-effector with a

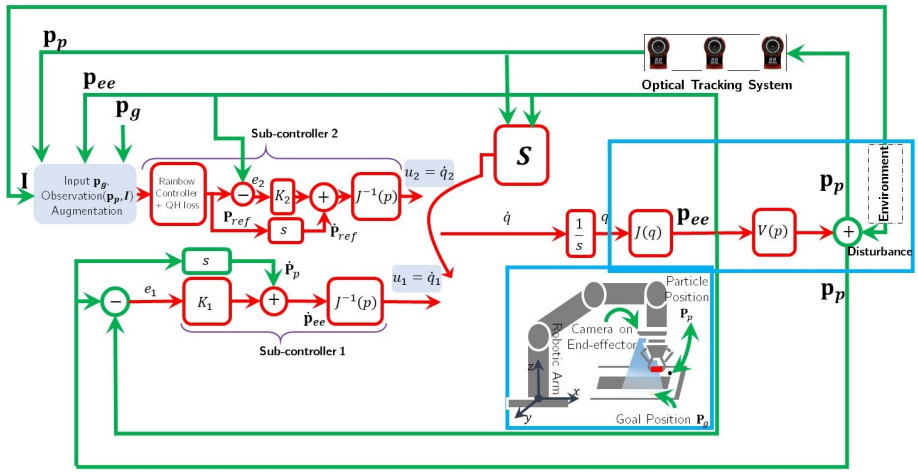


Figure 4.2: The switching controller architecture consists of Sub-controller 1 and Sub-controller 2 and a switching law ($\boldsymbol{S}$). $\boldsymbol{S}$ is a supervisory algorithm that controls the switching between the various controllers. The observations or feedback signals are: the magnetic particle position obtained through the optical tracking system ($\mathbf{P}_p$), partial information ($\mathbf{I}$) of the constrained environment captured by an RGB camera on the end-effector, i.e., The RGB camera only covers a portion of the environment for the Sub-controller 2. s represents derivative filter. In Sub-controller 1, K_1 represents a proportional gain for adjusting the asymptotic convergence of the controller. Joints are modeled as a single integrator which is represented by $\frac{1}{s}$. Inverse Jacobian and the kinematic of the robot is denoted by $J^{-1}(p)$ and $J(q)$, respectively. q and $\dot{q}$ represent joint positions and angular velocities. $V(p)$ models the unknown magnetic interaction dynamic between the external magnet on the end-effector and the particle with the input $\mathbf{P}_{ee}$ and output $\mathbf{P}_p$.

slight change concerning the end-effector's current position. Figure 4.4 shows details of the training.

As a reward function in the training of Sub-controller 2, each episode is terminated with a negative reward if there is no response for the inverse kinematic, the number of steps in each episode exceeds 150, or the end-effector would drive the particle into an obstacle. In addition, an episode is terminated with a positive reward if a goal is detected by the end-effector mounted camera within 2 cm in the xy-plane. It should be noted that throughout training, objectives are generated randomly during each episode.

The following is the switching law for enabling sub-controllers in the switching control architecture: To activate Sub-controller 2, the euclidean distance between the end-effector and particle locations must fall below a threshold of an arbitrary value, which is chosen to be 10 cm in this study.

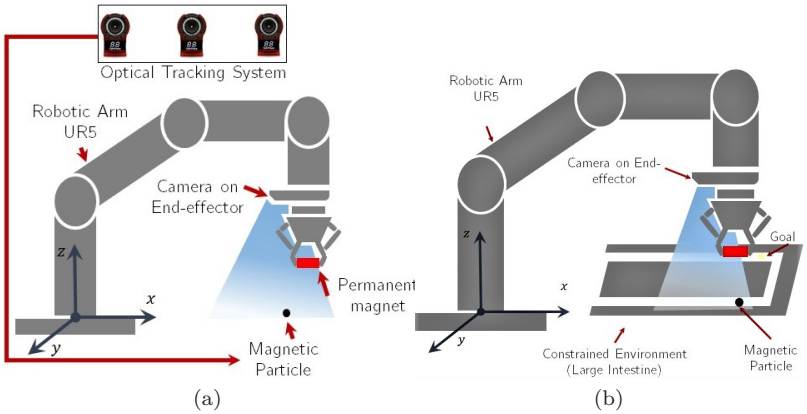


Figure 4.3: (a): The control-loop which employs analytical inverse kinematic solutions when Sub-controller 1 is activated: The feedback coming from the optical tracking system consists of the particle position and by employing the analytical inverse kinematic solutions, the robotic arm will locate the particle. (b): The control-loop mechanism which trains a customized Rainbow algorithm to navigate through a constrained workspace when Sub-controller 2 is activated: The feedback coming from the end-effector RGB camera consists of partial information about the environment.

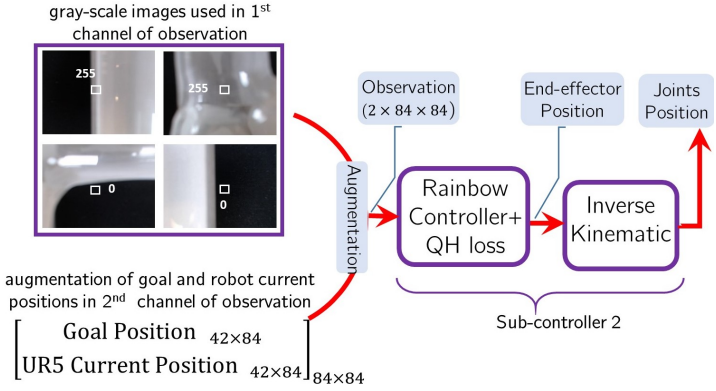


Figure 4.4: Observations for training Sub-controller 2: Two-channel images 84×84, i.e., $2 \times 84 \times 84$. The first channel includes a gray-scale image obtained from the end-effector RGB camera and the second channel is an augmentation of two matrices with the size 42×84 which are populated with the goal and robot current positions. The average of pixel values in the center of each gray-scale image is used for obstacle detection.

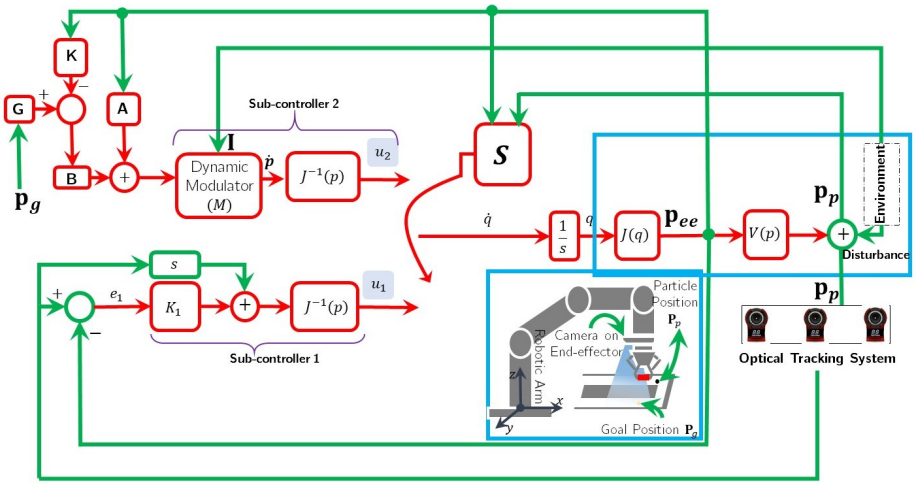


Figure 4.5: The switching controller architecture consists of Sub-controller 1 and Sub-controller 2, and a switching law ($\boldsymbol{S}$). $\boldsymbol{S}$ is a supervisory algorithm that controls the switching between the various controllers. The observations or feedback signals are: the magnetic particle position obtained through optical tracking system ($\mathbf{P}_p$), partial information ($\mathbf{I}$) of the constrained environment captured by an RGB camera on the end-effector, i.e., The RGB camera only covers a portion of the environment for the Sub-controller 2. $\mathbf{K}$ is the gain of the state feedback controller, $\mathbf{B}_{2\times2}$, and $\mathbf{A}_{2\times2}$ are the identity matrices ($Id_{2\times2}$). Furthermore, $\mathbf{G}_{2\times2}=[K\ Id]\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ Id & \mathbf{0}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{0}\\ Id\end{bmatrix}$. The symbol s represents derivative filter. In Sub-controller 1, K_1 represents a proportional gain for adjusting the asymptotic convergence of the controller. Joints are modeled as a single integrator which is represented by $\frac{1}{s}$. Inverse Jacobian and the kinematic of the robot is denoted by $J^{-1}(p)$ and $J(q)$, respectively. q and $\dot{q}$ represent joint positions and angular velocities. $V(p)$ models the unknown magnetic interaction dynamic between the external magnet on the end-effector and the particle with the input $\mathbf{P}_{ee}$ and output $\mathbf{P}_p$.

4.1.2.2 Attractor Dynamics-based Approach

In the Attractor Dynamics-based Method, a state feedback controller and a dynamic modulator [KB12, Section 3.2] are designed to ensure that the robot will not hit convex obstacles while carrying the particle toward a goal. The block diagram of the algorithm is shown in Figure 4.5. Sub-controller 1 and Switching Law are the same as what is discussed in Section 4.1.2.1. For Sub-controller 2:

$$\begin{aligned}\dot{p}(t) &= \mathbf{A}^{'}p(t) + \mathbf{B}^{'}u(t),\\ o(t) &= p(t)\end{aligned} \tag{4.1}$$

where $p(t) = \begin{bmatrix} x & y \end{bmatrix}^T \in \mathbb{R}^{2\times 1}$ is Cartesian position of the end-effector in a 2-D space (xy-plane), $\mathbf{B}'_{2\times 2}$, a state matrix and $\mathbf{A}'_{2\times 2}$, an input matrix, and $Id_{2\times 2}$ is the identity matrix. It is worth noting that $\mathbf{A}' = M\mathbf{A}$ and $\mathbf{B}' = M\mathbf{B}$. Furthermore, $u(t) \in \mathbb{R}^{2\times 1}$ is a 2-by-1 control input. Let P_g denotes the desired constant 2-by-1 vector for the output $o(t)$ to track asymptotically.

The control goal of Sub-controller 2 is to design a state feedback controller in which $u(t)$ depends on $p(t)$ and P_g so that the regulation error $e(t) = \mathrm{P}_g - o(t)$ goes to zero when $t \to \infty$. It can be shown ([Bro91; CLS04]) that the state feedback control law:

$$u(t) = \mathbf{G}\mathrm{P}_g - \mathbf{K}p(t) \tag{4.2}$$

makes System 4.1, a globally asymptotically stable system, where $\mathbf{G}_{2\times 2} = [K\ Id]_{2\times 4} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ Id & \mathbf{0} \end{bmatrix}^{-1}_{4\times 4} \begin{bmatrix} \mathbf{0} \\ Id \end{bmatrix}_{4\times 2}$ and $\mathbf{K}$ is the gain of the state feedback controller.

Although the control law 4.2 makes the manipulator stable, it cannot prevent the robot from colliding with obstacles. Therefore, a real-time obstacle avoidance strategy should be considered together with the described law. To design the obstacle avoidance protocol, we follow the same line of ideas as in [KB12]. We consider rectangular 2-D obstacles based on superellipse curves, where $\Gamma(p) : \|\frac{x-x_o}{a}\|^n + \|\frac{y-y_o}{b}\|^n = 1$ represents the boundary points of an obstacle with the center point $[x_o, y_o]$, furthermore, a and b are called the semi-diameters of the curve. In other words, the curve $\Gamma(p)$ is contained in the rectangles $\|x - x_o\| <= a$ and $\|y - y_o\| <= b$. The similarity of the curve to a rectangle is adjustable with the parameter n, and with $n > 2$, the curve looks like a rectangle with rounded corners.

The dynamic modulator matrix M (designed in [KB12, Section 3.2]) propagates the influence of the obstacle on the motion flow with the maximum effect at the boundaries of the obstacle and vanishes for points far from it. Applying the dynamic modulator to the linear system 4.1 yields:

$$\dot{p}(t) = M\Big(\mathbf{A}p(t) + \mathbf{B}u(t)\Big) \tag{4.3}$$

where $u(t)$ is given in Equation 4.2. Details of design, stability and convergence analysis of the system with a modulation matrix can be found in[KB12]. Here we skip the analysis for brevity.

For the computer implementation of 4.3, a simple way is to replace the derivative by a difference

$$p(t_{k+1}) = M\Big(\mathbf{A}p(t_k) + \mathbf{B}\mathbf{G}\mathrm{P}_g - \mathbf{K}p(t_k)\Big)h + p(t_k)h$$

where t_k is the sampling instant and $h = t_{k+1} - t_k$ is the sampling period.

4.1.2.3 ERRT-based Approach

To better contextualize the proposed RL-based control algorithm, we additionally propose employing an ERRT-based algorithm as a path planner within the Sub-controller 2, the specifics of which are detailed in [BV02].

ERRT is a sampling-based planning method for a continuous domain. Its two extensions compared to RRT method [LaV+98] —the waypoint cache and adaptive cost penalty search— enable the method to explore the environment and plan an obstacle-free path on the fly. However, the quality of the generated path and efficient replanning heavily depend on the processing of the received obstacle images from the environment. Therefore, to be able to use this algorithm to solve the presented problem, we develop a technique to optimally allocate obstacles in rectangular shapes to be used in the ERRT algorithm. In addition, the applicability of the algorithm in real-time heavily depends on how much the environment is polluted by obstacles, the complexity of obstacle shapes, whether obstacles are static or dynamic, and how fast obstacles are changing.

The block diagram of the switching controller based on this approach is presented in Figure 4.6. Sub-controller 1 and Switching Law are the same as what was mentioned in Section 4.1.2.1. However, in Sub-controller 2, ERRT, instead of the RL-based algorithm, is used as a path planner in Cartesian space, and the output is fed to the inverse kinematic function to generate joint angles.

In the obstacle processor, a quadtree [FB74] as a tree data structure is used for spatial 2-D searching to optimally partition obstacles into a minimum number of rectangular obstacles within a grayscale image, respecting a user-defined threshold for the height and width of the smallest acceptable rectangle, as depicted in Figure 4.7. As it is shown, there are regions with different shades of gray, and the contrast ranges from black —pixel value 0— at the weakest intensity to white —pixel value 255— at the strongest. To reduce the amount of computation in the ERRT algorithm, only regions with a color intensity of 50 or less are considered. These rectangular obstacles can be used efficiently by the ERRT method.

4.1.2.4 A Discussion on Stability

For the stability of this Sub-controller 1 in RL, Attractor dynamics, and ERRT-based methods, it can be shown that

$$\begin{aligned} e_1 &= \mathbf{P}_p - \mathbf{P}_{ee}, \\ \dot{e}_1 &= \dot{\mathbf{P}}_p - \dot{\mathbf{P}}_{ee}, \\ \dot{e}_1 &= \dot{\mathbf{P}}_p - J(q)\dot{q}, \\ \dot{e}_1 &= \dot{\mathbf{P}}_p - J(q)J^{-1}(q)\Big(\dot{\mathbf{P}}_p + K_1(\mathbf{P}_p - \mathbf{P}_{ee})\Big), \\ \dot{e}_1 &= -K_1 e_1 \end{aligned} \tag{4.4}$$

where K_1 is a 3-by-3 diagonal positive define matrix, then it is guaranteed that Cartesian vector-valued e_1 converges to 0 asymptotically for any initial values.

Remark 4.1.1. In the implementation of the derivative filter s within our control system, the choice was justified by several factors that ensure minimal noise impact. Primarily, the system benefited from a high signal-to-noise ratio (SNR) in a controlled environment, attributable to the use of decent imaging sensors and

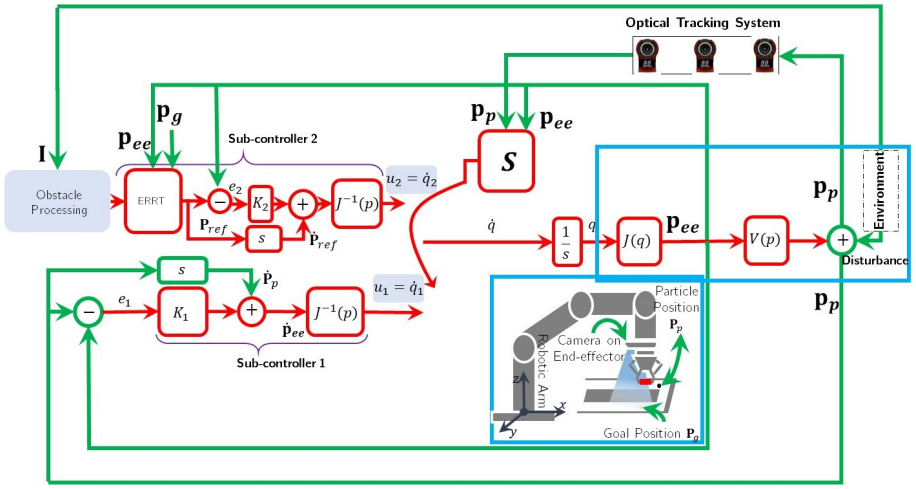
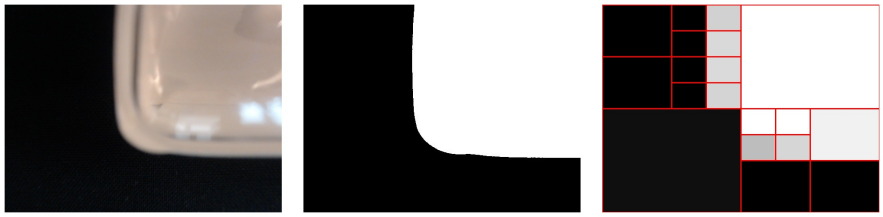


Figure 4.6: The switching controller architecture consists of Sub-controller 1 and Sub-controller 2, and a switching law ($\boldsymbol{S}$). $\boldsymbol{S}$ is a supervisory algorithm that controls the switching between the various controllers. The observations or feedback signals are: the magnetic particle position obtained through an optical tracking system ($\mathbf{P}_p$), partial information ($\mathbf{I}$) of the constrained environment captured by an RGB camera on the end-effector, i.e., the RGB camera only covers a portion of the environment for the Sub-controller 2. $\mathbf{I}$ is fed to the obstacle processing unit, which optimally partitions obstacles into a minimum number of rectangular obstacles. The ERRT unit generates an obstacle-free path based on the given feedback information. The symbol s represents a derivative filter. In Sub-controller 1, K_1 represents a proportional gain for adjusting the asymptotic convergence of the controller, and in Sub-controller 2, K_2 represents a positive, definite 3-by-3 matrix for adjusting the asymptotic convergence of the controller. K_2 is a simple model for the internal control of the robot. Joints are modeled as a single integrator, which is represented by $\frac{1}{s}$. Inverse Jacobian and the kinematic of the robot are denoted by $J^{-1}(p)$ and $J(q)$, respectively. q and $\dot{q}$ represent joint positions and angular velocities. $V(p)$ models the unknown magnetic interaction dynamic between the external magnet on the end-effector and the particle with the input $\mathbf{P}_{ee}$ and output $\mathbf{P}_p$.

(a)



(b)

Figure 4.7: Inputs and outputs for the obstacle processing unit. (a)-(b): the most left images are the inputs to the unit. The input images are converted to grayscale images first, and then a quadtree algorithm is used to optimally partition obstacles within a grayscale image into a minimum number of rectangular obstacles respecting an user-defined threshold for the height and width of the smallest acceptable rectangle. In these figures, the threshold is 50 for both height and width.

their refined data acquisition protocols, which inherently suppressed noise levels significantly. Moreover, we employed a robust numerical method—the mean filter—for differentiation, which is specifically designed to attenuate noise while accurately estimating derivatives. Empirical validation also underpinned our approach: the experimental trials consistently demonstrated that the system's performance remained well within acceptable limits, reinforcing the validity of using s. Data collection was conducted in a controlled environment, which greatly diminished the influence of external noise variables. Collectively, these measures effectively mitigated noise, justifying the derivative filter's use in the system's design. Nonetheless, it is recommended to reassess the application of s when transitioning the control system to practical, real-world environments.

Remark 4.1.2. Incorporating more realistic joint dynamics, such as friction, alters the stability analysis. The simple integrator model assumes ideal, friction-less motion, leading to a straightforward relationship between joint velocities and positions. However, when friction is considered, it introduces a non-linear, dissipative force that opposes motion. Friction is naturally considered velocity-dependent, altering the stability analysis. Specifically, friction could aid in

damping oscillations and stabilizing the system, but it also introduce non-linear behaviors that complicate the convergence of e_1 to zero. The presence of friction means that the system's response to control inputs becomes less predictable and more sensitive to the specific characteristics of the friction model used, such as static vs. dynamic friction, and the friction coefficient. As a result, the controller might need to be more sophisticated, possibly requiring adaptive or robust or adaptive robust control strategies to ensure stability and desired performance.

As it is discussed in Section 4.1.2.1, a closed control architecture within industrial robots or cobots (collaborative robots) such as UR5 has the responsibility of regulating joint angles to desired values (sometimes for simplicity, each joint of an industrial arm might be seen as a simple integrator, which is being controlled internally). For the stability of this Sub-controller 2 in RL and ERRT-based methods, it can be shown that

$$\begin{aligned} e_2 &= \mathbf{P}_{ref} - \mathbf{P}_{ee}, \\ \dot{e}_2 &= \dot{\mathbf{P}}_{ref} - \dot{\mathbf{P}}_{ee}, \\ \dot{e}_2 &= \dot{\mathbf{P}}_{ref} - J(q)\dot{q}, \\ \dot{e}_2 &= \dot{\mathbf{P}}_{ref} - J(q)J^{-1}(q)\Big(\dot{\mathbf{P}}_p + K_1(\mathbf{P}_{ref} - \mathbf{P}_{ee})\Big), \\ \dot{e}_2 &= -K_2 e_2 \end{aligned} \tag{4.5}$$

where K_2 is a 3-by-3 diagonal positive, define matrix, then it is guaranteed that Cartesian vector-valued e_2 converges to 0 asymptotically for any initial values.

Both RL and ERRT algorithms act as path planners, and it can be seen that the robot's internal controller stabilizes the system as long as an obstacle-free path within the workspace of the robot is generated. This is determined by how well Sub-controller 2 is trained in the RL algorithm or how good the quality of a path is in the ERRT method. Furthermore, in the Attractor dynamic-based method, the global asymptotic stability of Sub-controller 2 is guaranteed, as it is discussed earlier in Section 4.1.2.2.

It should be noted that each control loop has different set-points, so the loops are entirely separate in that sense. In other words, switching between those two stable control loops does not introduce instability to the system, and the loops accomplish separate sub-tasks with different set-points. Therefore, this may raise the question of whether or not there is a possibility of having only one controller to accomplish the task. Since there is no need to have a collision avoidance strategy when the manipulator is moving to the location of the disturbed particle, two different controllers are considered.

4.1.3 Simulations and Experiments

We employ the Universal Robots UR5 robotic arm for simulations and experiments. The UR5 manipulator has a gripper that holds a neodymium block magnet in place. In Figure 4.8, images from the end-effector camera are utilized to detect the magnetic particle and identify barriers in the surroundings.

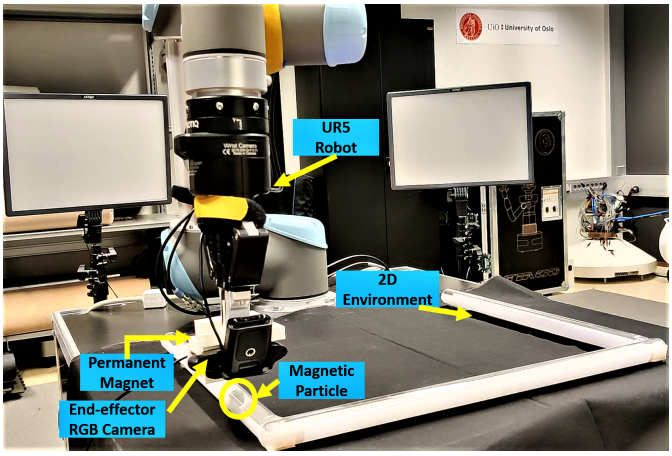


Figure 4.8: The experimental setup: UR5, Permanent Magnet and RGB camera mounted on the end-effector, the 2D constrained workspace, and the object (sphere magnet).

We use a sphere of NdFeB magnet with a grade of N42 and a diameter of 5 mm as a magnetic particle. Besides, a neodymium block magnet with dimensions $50 \times 25 \times 10$ mm and a grade of N35 is attached to the UR5's end-effector to produce enough magnetic field strength to carry the magnetic particle. To prevent two magnets from colliding, the UR5 moves in an imaginary plane 10 cm above the table's surface.

4.1.3.1 Simulations, Experiments, and Results: Reinforcement Learning-based Approach

Training for a policy in the real world can be costly. Simulations can speed up the learning process and help avoid potentially unwanted actions that can damage the robot or the surrounding environment. OpenAI Gym and PyBullet [CB19] are used for this purpose. However, modeling a complex environment or robots can be challenging, and this may introduce a simulation-reality gap [Bou+18]. To minimize this gap, the UR5 swept and imaged the whole real-world environment using the mounted RGB camera (with steps of 1 cm in both x and y-axes). As discussed, the UR5 end-effector was kept in an imaginary plane at a 10 cm distance from the surface of the table during the imaging process.

It is worth mentioning that in the simulation, the magnetic particle is not considered, and therefore the whole switching control architecture is not implemented. After training, the average success rate for the learned policy is 100% for randomly generated initial and goal positions (calculated over 30 random restarts).

Afterward, we consider the magnetic particle in the loop, and the full controller structure is implemented. In the real world, the magnetic particle is

consistently carried to a randomly placed target, starting from a random position. We repeated undisturbed scenarios for arbitrary initial and goal positions for 30 episodes, and the average success rate is 100% with a mean tracking error of 0.02 (m) in both axes and an average maximum error of 0.08 (m) and 0.09 (m) in the x and y-axes, respectively.

Finally, the robustness of the system to deal with external disturbances that could result from real-world uncertainties is studied. As disturbances occur, we physically draw the magnetic particle away from the magnetic control zone during active control. We repeated the disturbed scenarios for random disturbances, initial positions, and the target positions for 30 episodes, and the average success rate was 96.6%.

Remark 4.1.3. Sub-controller 1 employs kinematic control; therefore, its performance is satisfactory only when the introduced disturbances are neither too rapid nor require significant accelerations. Furthermore, due to the lack of control over the magnetic field's strength, the robot may struggle to navigate the particle through corners in the environment. Essentially, a failure occurs when the end-effector would drive the particle into an obstacle, there is no response for the inverse kinematic, or the number of steps in each episode exceeds 150.

4.1.3.2 Simulations and Results: Attractor Dynamics-based Approach

The following example illustrates the Attractor Dynamic-based method without any disturbance applied.

Example 4.1.4. Consider a 2-D shape of the constrained environment with the manipulator's initial Cartesian position $[-0.78, 0.73, 0.10]$ and the target position $[0.77, 0.19, 0.10]$. An overview of the simulation is shown in Figure 4.9. It should be noted that the manipulator can partially sense an obstacle with a maximum height and width of 2 cm. The system employs the state feedback control law given in Equation (4.2). Since there is no perturbation, Sub-controller 1 (as shown in Figure 4.5) is not active.

An advantage of the considered conventional method is that there is no need for offline training or parameter tuning; however, the method cannot handle concave or connected obstacles. Since the method requires analytical modeling of the obstacle's boundary, we consider a rectangle around the part of the obstacle detected by the robot's sensor — we fit the point cloud with a rectangle—. However, the actual shape of the obstacle(s) may not necessarily be a rectangle. In other words, the approach might be conservative in some scenarios.

In the following examples, a disturbance is introduced to the motion of the particle.

Example 4.1.5. Consider a 2-D shape of a constrained environment shown in Figure 4.10. The manipulator's initial Cartesian position $[0.72, 0.76, 0.10]$ and the target position $[-0.77, 0.12, 0.10]$. A perturbation in x-axis with magnitude 0.63 m is applied to the particle when the particle is at the position $[-0.21, 0.55, 0]$ or at step number 50 where Sub-controller 1 (as shown Figure 4.5) is active and

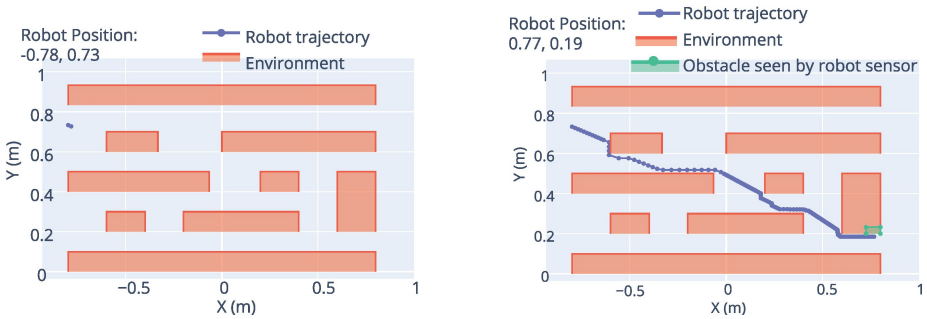


Figure 4.9: Online obstacle avoidance of UR5 carrying an undisturbed particle through a 2-D environment in Example 4.1.4. Information from a part of obstacle seen by the robot is used to make the particle goes around the obstacle. Since there is no perturbation is assumed, Sub-controller 2 carries the particle toward the target position.

takes the manipulator to the new position of the particle. When the manipulator is in place, Sub-controller 2 has the responsibility of carrying the particle toward the target position. Also in this example, the manipulator can partially sense an obstacle with a maximum height and width of 2 cm. An overview of the simulation is shown in Figure 4.10.

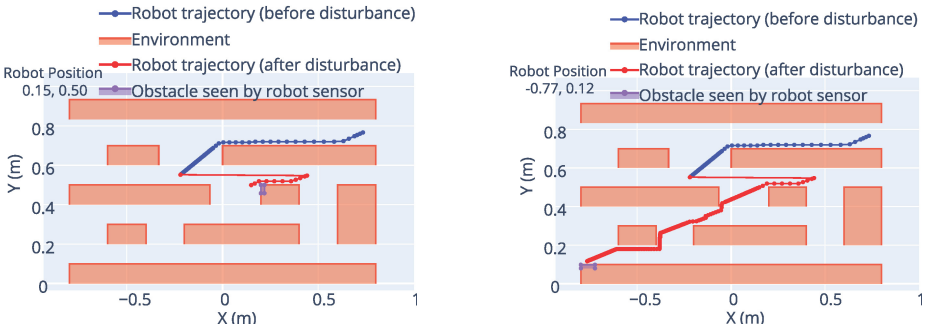
4.1.3.3 Simulations and Results: ERRT-based Approach

The following example illustrates the performance of the ERRT-based method with disturbances applied to the particle.

Example 4.1.6. Consider a 2-D environment as depicted in Figure 4.11 with the manipulator's initial Cartesian position $[-0.24, -0.27, 0.10]$ and the target position $[0.19, -0.28, 0.10]$.

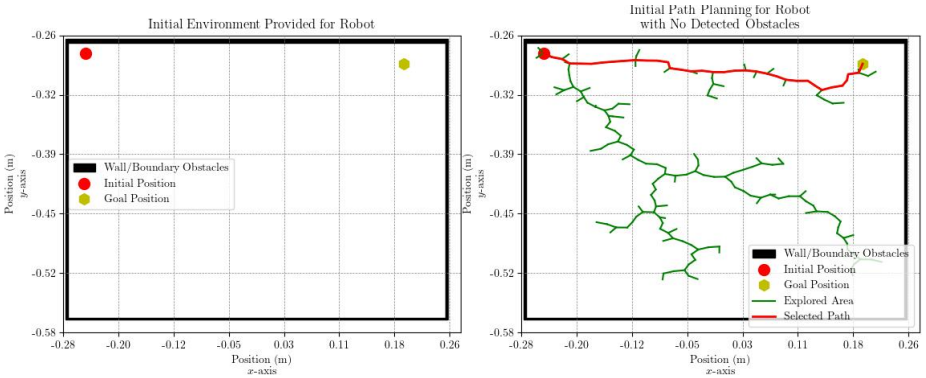
Two disturbances happen at locations $[0.21, -0.34, 0.10]$ and $[0.20, -0.47, 0.10]$ where the particle is displace to new positions $[-0.19, -0.48, 0.10]$ and $[-0.24, -0.28, 0.10]$, respectively, as shown in Figure 4.12 where initial, target, and disturbance positions, explored area, and selected path are depicted. In addition, it should be noted that the ERRT algorithm senses obstacles through received images from the environment depending on the robot's end-effector's current position (obstacles are partially seen by the camera mounted on the robot).

Figure 4.13 depicts the particle and end-effector trajectories together with Tracking Error (T.E.) in both axes x and y and the mean of T.E. through the environment in the presence of disturbances where the particle is supposed to reach the goal position $[0.21, -0.40, 0.10]$.



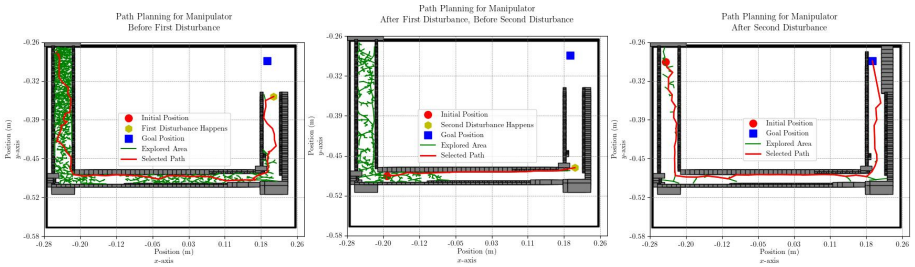
(a) (b)

Figure 4.10: Online obstacle avoidance of UR5 carrying a disturbed particle through a 2-D environment in Example 4.1.5. A disturbance in x-axis with magnitude 0.63 m is applied to the particle when the particle is at the position $[-0.21, 0.55, 0]$, as shown in Figure 4.10a. When the disturbance occurs, Sub-controller 1 takes the manipulator to the new position of particle. When the manipulators in the place, Sub-controller 2 has the responsibility of carrying the particle toward the target position (the path shown in red). Information from a part of the obstacle seen by the robot is used to make the particle goes around the obstacle.



(a) (b)

Figure 4.11: (a): Initial environment provided for robot with boundary obstacles to limit the search space. (b): Initial path planning by robot from the initial position to the goal robot with explored and selected paths and boundary obstacles to limit the search space. No obstacle is detected by the camera.



(a) (b) (c)

Figure 4.12: Initial, target, and disturbance locations, explored area and selected path. (a): Initial position before the first disturbance at $[-0.24, -0.27, 0.10]$, the target position at $[0.19, -0.28, 0.1]$, and the first disturbance occurs at $[0.21, -0.34, 0.1]$. (b): Initial position after the first disturbance at $[-0.19, -0.48, 0.1]$, the target position is the same as before, and the second disturbance occurs at $[0.20, -0.47, 0.1]$. (c): Initial position after the second disturbance at $[-0.24, -0.28, 0.1]$, the target position is kept the same as before, and no disturbance occurs.

4.1.4 Discussions

We examined the feasibility of implementing the Rainbow algorithm [Hes+17] to accomplish set-point regulation tasks within a constrained environment in the presence of disturbances. The results suggest that deep RL methods as a trajectory planner prevent the robot from encountering singularities and guide the particle toward a goal while avoiding obstacles and accommodating for disturbances. Furthermore, the results show that it is possible to train the RL algorithm fully within a simulation environment and deploy it as-is in a real-world scenario for remote magnetic control with reliable behavior in the presence of disturbances and uncertainties. Although traditional control approaches may ensure some performance levels under restricted assumptions, their design is challenging, especially when only partial information from the environment is available at each time step. Also, in many traditional path planning and control design approaches, an exact analytical model of the robot and its interaction with the surrounding environment is a necessity; contrastingly, RL methods do not rely on pre-existing knowledge or models.

Compared to traditional methods, the strength of RL-based methods is that they are applicable to the more complex environment without a need to re-design the controller structure, —which is based on neural networks—. Yet, there might be a need for more samples for offline training and proper tweaking of the controller's hyperparameters. We also experienced that learning performance is sensitive to the choice and format of observation space; therefore, learning parameters must be chosen carefully.

It is worth mentioning that the presented Attractor Dynamics-based

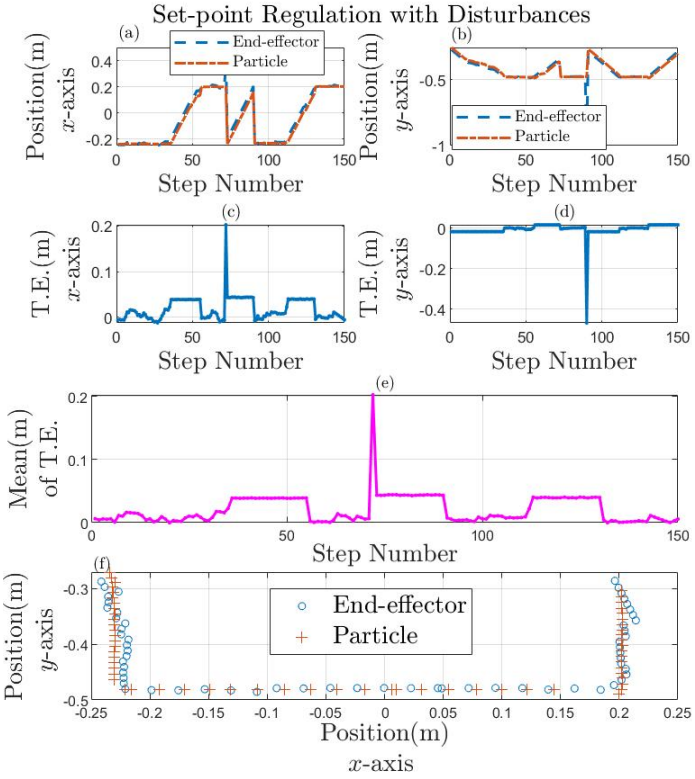


Figure 4.13: Set-point regulation with three disturbances in one episode. Disturbances happen at step numbers 72 and 90 (a,b): Cartesian positions of the end-effector and the magnetic particle are depicted in both x and y-axes. In (a), disturbance at step numbers 72, and in (b), a disturbance at step number 90 are visible. (c,d): Tracking Errors (T.E.) in each axis are shown. Errors in each axis increase when there is motion in that axes or when a disturbance happens. The magnitude of disturbances at step numbers 72 and step number 90 can be seen in (c) and (d), respectively. (e): Euclidean norm of errors in both axes at each step number is shown. (f): Locus of the particle and end-effector in the 2-D workspace is depicted.

conventional method can only be used where obstacle(s) are convex and are not connected. The method requires analytical modeling of the obstacle's boundary. To this end, we consider a rectangle around the part of the obstacle detected by the robot's sensor — we fit the point cloud with a rectangle—. However, the actual shape of the obstacle(s) may not necessarily be a rectangle. In other words, the approach might be conservative in some scenarios. Also, integration errors can cause issues in the discrete implementation of the approach in a way that the next computed point falls inside an obstacle boundary due to the integration error, and trajectories tend to stay inside the obstacle (as no trajectory can enter or leave obstacles).

To highlight the benefits of the proposed RL-based algorithm, an ERRT-based approach is also employed to accomplish the same task. ERRT is a sampling-based planning method that explores the environment and plans an obstacle-free path on the fly. In order to handle feeding the algorithm with varying obstacle shapes, a technique is developed to optimally partition obstacles into rectangular shapes that can be processed by the ERRT algorithm. This is the first difference between RL and ERRT-based methods. In the RL-based approach, there is no need to be concerned about the shape of obstacles as the whole obstacle image will be taken into account. Therefore, by approximating obstacles, the available workspace for generating an obstacle-free path might be reduced. At the same time, this approximation will also increase the computation loads to some extent, which eventually will lead to decreased bandwidth for the control loop.

In the proposed RL-based algorithm, training or exploring the environment takes place offline, yet in the ERRT method, environment exploration happens on the fly, and this directly affects whether the method can be used in real-time. If the environment is filled with complex and/or dynamic obstacles, the ERRT algorithm might not be responsive to the changes due to computational loads. However, this is not the case in the RL-based algorithm. In the presented scenario in this chapter, the environment is static, so employing the ERRT method was feasible.

In the ERRT method, there is a need for calibration of the environment with respect to the robot to keep track of the obstacle positions. Yet, in the RL-based approach, the current pose of the robot will be assigned to the received obstacle image, and this will be preserved in the neural network, which will enable employing the RL-based method in dynamic 3-dimensional environments without being concerned about the calibration. Still we may use ERRT method in 3-dimensional environments if we have a reliable way of mapping obstacles to the robot's reference frame.

4.2 Optimal Kinematic-based Controller for Magnetic Guidewire

We employ a simplified integration scheme presented in Section 3.3 to model a guidewire and forces resulting from the interference of magnetic fields to provide a rapid model reconstruction. Furthermore, using the presented model and by

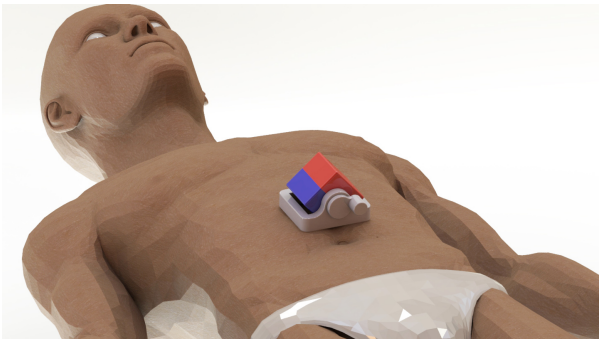


Figure 4.14: Proposed steering setup including an external magnet

orienting the external magnetic field, a guidewire is steered into a constrained environment. Finally, simulations illustrate the approach performance on a soft rod where an external magnetic field is oriented to form the desired shape for a continuum rod and steer it within a vascular-like environment.

4.2.1 Steering Setup

A single external magnet is employed in this study to introduce forces to the tip for steering purposes. The proposed setup is shown in Figure 4.14. Furthermore, the external magnet is modeled as a set of 2000 dipoles rotating around the y-axis and Figure 3.15 shows the resulting flux, the applied force on the tip dipole, and the projected force on the spline.

The external magnet is located at a 7 cm distance from the GI tract. The idea is to rotate the external magnet to steer the guidewire toward a desired location inside a GI tract structure, as shown in Figure 3.16. Length and diameter of GI tract structure are 8×10^{-1} m and 1.5×10^{-2} m. The tip embedded neodymium magnet is a cylindrical magnet with a diameter of 2 mm, height of 3 mm, weight of 7.2×10^{-5} kg, and residual magnetism of 1.37 T. A circular cross-section guidewire with the radius 2.5×10^{-3}, Young modulus constants 550 and 600 KPa for rotational and linear springs are considered, i.e., the rotational and linear spring constants are 10.79 Nm^{-1} and 11.78 Nm^{-1}, respectively. For the external magnet, a block magnet 50.8 mm $\times$ 50.8 mm $\times$ 50.8 mm with residual magnetism of 1.3 T is considered. Furthermore, it should be noted that as boundary conditions, the guidewire is always locked to the direction of the GI tract, i.e., tangent to the GI tract, and cannot bungle up inside the environment, as it has been mentioned in the last step in Algorithm 1.

Remark 4.2.1. The choice of 2000 dipoles for the modeling of the external magnet to steer a guidewire within a GI tract strikes a balance between modeling accuracy and computational efficiency. This number provides sufficient detail to simulate the magnetic field, especially at the guidewire's tip for steering (considering the magnetic distance from the GI tract, Young modulus and spring constants etc.). Simultaneously, it represents a point where the computational

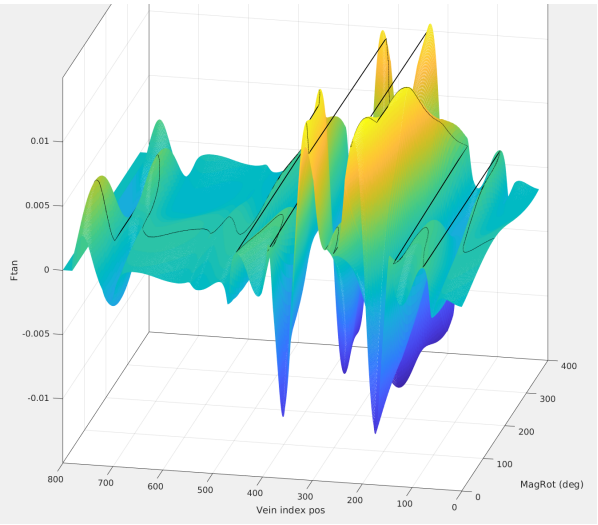


Figure 4.15: Projected force (N) is applied to the guidewire tip dipole alongside the GI tract resulting from the orientation of the external magnet with respect to the rotation of the external magnet at different locations of the GI tract, i.e., the GI tract index positions. The maximum projected force applied to the dipole embedded at the tip of the guidewire is shown by the solid black curve.

demands of the simulation remain manageable, aligning with a specific precision requirement of the application.

4.2.2 Simulation Results

For the simulation, the external magnet only has a rotational movement. Figure 4.15 shows projected forces tangent to the GI tract structure obtained from different rotation angles. The maximum projected force on the spline curve resulting from a specific rotation angle is depicted in Figure 4.16. In other words, Figure 4.16 depicts the maximum projected force applied to the dipole embedded at the tip of the guidewire resulting from a specific rotation angle of the external magnet at each position of the GI tract.

4.3 Findings and Implications

In this chapter, we mainly discussed deriving and validating a customized Rainbow RL method for online trajectory planning and remote control of a ferromagnetic particle. Using magnetic actuation, the robot learned to robustly carry a small ferromagnetic object in a constrained environment. Furthermore, the trained network is integrated with the two-module controller, which is deployed as-is in a validation experiment in the real world, where the experiment showed the robustness of the approach against disturbances.

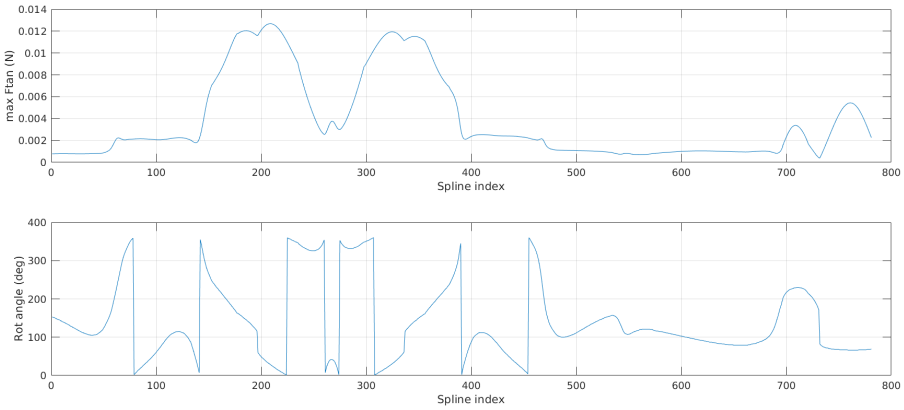


Figure 4.16: The maximum projected force applied to the dipole embedded at the tip of the guidewire results from a specific rotation angle of the external magnet at each position of the GI tract.

Also, a conventional controller based on the Attractor Dynamics-based approach is designed. Afterward, simulations were carried out, and the shortcomings of the proposed method were also discussed. Finally, the experiment results from the ERRT-based method highlight the improved robustness to dynamic environments offered by the proposed RL-based algorithm. For future work, a three-dimensional environment can be investigated. Also, the constrained environment is static, meaning it does not change over time. However, by training the RL algorithm in dynamic environments, there is a possibility of considering time-varying workspaces.

In the second section of this chapter, the possibility of steering a guidewire inside a complex environment by rotating an external magnet is demonstrated. The results open up possibilities to construct a rapid model and, therefore, model-based control algorithms for magnetically-actuated continuum manipulators for closed-loop control applications to reduce the reality gap. Clearly, further research will be needed to validate the results. Therefore, the next stage of this research will be the experimental confirmation of the presented framework and considering more external magnets as actuators for shape formation of CMs in difficult-to-reach environments.

References

[Ami+17] Amin, F. U. et al. “Osmotin-loaded magnetic nanoparticles with electromagnetic guidance for the treatment of Alzheimer’s disease”. In: *Nanoscale* vol. 9, no. 30 (2017), pp. 10619–10632.

[BDM17] Bellemare, M. G., Dabney, W., and Munos, R. “A distributional perspective on reinforcement learning”. In: *arXiv preprint arXiv:1707.06887* (2017).

[Bou+18] Bousmalis, K. et al. “Using simulation and domain adaptation to improve efficiency of deep robotic grasping”. In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 4243–4250.

[Bou12] Bourzac, K. “Nanotechnology: carrying drugs”. In: *Nature* vol. 491, no. 7425 (2012), S58–S60.

[Bro91] Brogan, W. L. *Modern control theory*. Pearson education india, 1991.

[BV02] Bruce, J. and Veloso, M. M. “Real-time randomized path planning for robot navigation”. In: *Robot soccer world cup*. Springer. 2002, pp. 288–295.

[CB19] Coumans, E. and Bai, Y. *PyBullet, a Python module for physics simulation for games, robotics and machine learning*. pybullet.org. 2016–2019.

[CLS04] Chen, B. M., Lin, Z., and Shamash, Y. *Linear Systems Theory*. 2004.

[Dab+18] Dabney, W. et al. “Implicit quantile networks for distributional reinforcement learning”. In: *International conference on machine learning*. PMLR. 2018, pp. 1096–1105.

[FB74] Finkel, R. A. and Bentley, J. L. “Quad trees a data structure for retrieval on composite keys”. In: *Acta informatica* vol. 4, no. 1 (1974), pp. 1–9.

[For+17] Fortunato, M. et al. “Noisy networks for exploration”. In: *arXiv preprint arXiv:1706.10295* (2017).

[He+16] He, K. et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[Hes+17] Hessel, M. et al. “Rainbow: Combining improvements in deep reinforcement learning”. In: *arXiv preprint arXiv:1710.02298* (2017).

[Hos+17a] Hoshiar, A. K. et al. “Studies of aggregated nanoparticles steering during magnetic-guided drug delivery in the blood vessels”. In: *Journal of Magnetism and Magnetic Materials* vol. 427 (2017), pp. 181–187.

[Hos+17b] Hoshiar, A. K. et al. “Studies of aggregated nanoparticles steering during magnetic-guided drug delivery in the blood vessels”. In: *Journal of Magnetism and Magnetic Materials* vol. 427 (2017), pp. 181–187.

[Hos+18] Hoshiar, A. K. et al. “A novel magnetic actuation scheme to disaggregate nanoparticles and enhance passage across the blood–brain barrier”. In: *Nanomaterials* vol. 8, no. 1 (2018), p. 3.

[KB12] Khansari-Zadeh, S. M. and Billard, A. “A dynamical system approach to realtime obstacle avoidance”. In: *Autonomous Robots* vol. 32, no. 4 (2012), pp. 433–454.

[LaV+98] LaValle, S. M. et al. “Rapidly-exploring random trees: A new tool for path planning”. In: (1998).

[Lu+18] Lu, H. et al. “A bioinspired multilegged soft millirobot that functions in both dry and wet conditions”. In: *Nature communications* vol. 9, no. 1 (2018), pp. 1–7.

[Mni+13] Mnih, V. et al. “Playing atari with deep reinforcement learning”. In: *arXiv preprint arXiv:1312.5602* (2013).

[Mni+15] Mnih, V. et al. “Human-level control through deep reinforcement learning”. In: *nature* vol. 518, no. 7540 (2015), pp. 529–533.

[OC20] Obando-Ceron, J. S. and Castro, P. S. “Revisiting Rainbow: Promoting more insightful and inclusive deep reinforcement learning research”. In: *arXiv preprint arXiv:2011.14826* (2020).

[Sch+15] Schaul, T. et al. “Prioritized experience replay”. In: *arXiv preprint arXiv:1511.05952* (2015).

[Sch+19] Schuerle, S. et al. “Synthetic and living micropropellers for convection-enhanced nanoparticle transport”. In: *Science advances* vol. 5, no. 4 (2019), eaav4803.

[Tiw+12] Tiwari, G. et al. “Drug delivery systems: An updated review”. In: *International journal of pharmaceutical investigation* vol. 2, no. 1 (2012), p. 2.

[VGS15] Van Hasselt, H., Guez, A., and Silver, D. “Deep reinforcement learning with double q-learning”. In: *arXiv preprint arXiv:1509.06461* (2015).

[VS16] Vikram Singh, A. and Sitti, M. “Targeted drug delivery and imaging using mobile milli/microrobots: A promising future towards theranostic pharmaceutical design”. In: *Current pharmaceutical design* vol. 22, no. 11 (2016), pp. 1418–1428.

[Wan+16] Wang, Z. et al. “Dueling network architectures for deep reinforcement learning”. In: *International conference on machine learning*. 2016, pp. 1995–2003.

[WD92] Watkins, C. J. and Dayan, P. “Q-learning”. In: *Machine learning* vol. 8, no. 3-4 (1992), pp. 279–292.

[WJL15] Wen, H., Jung, H., and Li, X. “Drug delivery approaches in addressing clinical pharmacology-related issues: opportunities and challenges”. In: *The AAPS journal* vol. 17, no. 6 (2015), pp. 1327–1340.

Chapter 5

Discussion

The work presented in the thesis will be discussed in this chapter. Initially, we examine the motives behind the utilized and developed modeling and control methods. After that, the chapter concludes with a discussion on the results, performance, and limitations of the research in the thesis.

The Nature of Models and the Quest for Illumination

Many approaches have been devised by researchers to tackle the modeling of a CM. "All models are wrong as they are approximations and therefore don't show the whole truth, but some are useful. The main question of interest is: Is the model illuminating and useful?" [Box76]. At the same time, "we must recognize that eternal truth is not within our grasp" [NW72]. "The idea that complex physical, biological, or sociological systems can be exactly described by a few formulae is patently absurd" [Cox90]. In other words, "there is no formula that can deliver all truth, all harmony, all simplicity. No theory of everything can ever provide total insight, for to see through everything would leave us seeing nothing at all" [Bar08]. Thus, we strived to figure out how we might get the model as close to a real system as possible, at least in some aspects, in order to address the issue of obtaining an illuminating model of complex physical systems like soft CMs.

Lie Group Variational Integration and its Challenges

Focusing on symplectic integration[1]of Hamiltonian systems guided us to investigate Lie group variational integration scheme to model CMs. The solver[2], presented in the first paper, is not only symplectic and conserves system invariants such as energy and momentum maps for long-time simulations, but also its unique formulation provides the possibility of being implemented in parallel. The latter is surprisingly nontrivial, meaning that each element in the mesh can possibly have a different time step and does not need to be updated in a consecutive manner. The first article does not investigate the parallel implementation of the solver, and the solver results in average absolute and maximum errors in tip position estimation of 0.33 mm and 1.40 mm, respectively, for a manipulator length of 85.5 mm in quasi-static experiments[3]. In those experiments, the manipulators experienced large deformations and external loads; nevertheless,

[1]See Appendix A

[2]The term solver refers to a numerical algorithm or computational method that solves the underlying equations governing the system's behavior.

[3]See Table I.8 for more details.

the worst-case errors were less than 2% of the manipulators' length. It is also worth noting that Lie group variational based solvers for geometrically exact modeling of soft CMs have not been widely used in the robotics community due to a variety of factors, including difficult implementation and exploitation for control purposes. Furthermore, it should be pointed out that the provided integration scheme is most suited to CMs where external wrenches and loads are applied at actuation points by electric[Yun+12], thermal[Hag+10; LK04], or magnetic[Kim+19] actuation mechanisms. In CMs where wrenches are conveyed along the bodies of manipulators to their tips via hydraulic[IIS02], pneumatic[RWM18], or tendon-deriven[Ren+12] actuations, the solver might be challenging to deploy.

Neural Network-Based Modeling for Improved Real-time Performance

To address the shortcoming of the integration scheme, especially its real-time performance, some further studies were carried out in Paper II and Paper IV. The underlying principles and structure of the modeling method —a parallelizable neural-network-based prediction— are heavily inspired by the formulation of the solver presented in the Paper I and the fact that the solver, to a very good extent, mimics the true behavior of CMs. While trained neural network-based prediction algorithms on data-sets may be more likely to pick up more on the structure of the data than the underlying dynamics of CMs, tests performed in Section II.4, have demonstrated that trained models can be generalized when new input profiles are introduced. High real-time bandwidth for the presented algorithm is shown through several experiments on two different CMs, i.e., a magnetically-actuated soft CM fabricated from a urethane rubber Polymer Matrix Composite and a commercial tendon-driven transesophageal echocardiography probe. It might be worth stating that the neural-network-based model is straightforward in applying to tendon-driven CMs without being concerned with how wrenches are being propagated through the CM to its tip.

Better estimation quality is tied to having a comprehensive data-set, and this means that obtaining such data-sets might be costly in the real world, and it may not always be feasible to build such data-sets using high-fidelity classical models. A closer look at the PDEs governing the dynamics of CMs whose high order introduces a level of complexity such that constructing physics-aware numerical solutions has not been adequately investigated so far; however, there are numerous attempts that consider more basic and simpler types of PDEs utilizing neural networks. Murray Gell-Mann, a Nobel Prize-winning physicist, nicely puts it: "Someone should be studying the whole system, however crudely that has to be done, because no gluing together of partial studies of a complex nonlinear system can give a good idea of the behavior of the whole".

Exploring Control Methods: Reinforcement Learning and Classical Approaches

In the next step of this thesis, we explored more extensively the topic of CM control, zeroing in on magnetically actuated CMs, the control of which essentially comes down to the control of actuation points. In other words, the very first step in comprehending how a CM may be controlled in confined, dynamic environments is learning how the actuation point of the CM, which can be thought of as a small, rigid particle, can be controlled in such workspaces. Hence, Paper III tackled the issue of controlling a small particle in a constrained environment using an RL algorithm. RL algorithms are organic choices for controlling the behavior of small-scale particles in a general constrained workspace, as they may be more robust in dealing with uncertainties and also less difficult to design than traditional approaches, where they typically require more restricting assumptions on the workspaces or prior knowledge about the environment. It should be noted, however, that the control performance may not be generalizable to the case in which the same method is used for a CM, as the soft portions of the manipulator undergo elastic deformation, which introduces effects that may be incorporated as unknown three-dimensional disturbances on the actuation points.

Although three approaches were evaluated, each has advantages and disadvantages. In other words, classical control systems may be able to ensure some performance levels under restricted assumptions, but their design is challenging, particularly when only partial information from the environment is accessible at each time step. In addition, many conventional path planning and control design methods require an exact analytical model of the robot and its interaction with the surrounding environment, whereas RL methods do not rely on prior information or models. Implementing RL algorithms for real-world robotic applications can be costly, yet this can be mitigated to some extent by training algorithms in simulation environments and deploying them as-is in real-world scenarios.

Applying Developed Models for Control in Constrained Environments

A particularly important and relevant point that should be discussed here is how the models developed in the two first publications might be applied to control a CM in confined, dynamic environments such as the intervascular system or the large intestine of the human body. A requirement to deploy those models is the need to know the exerted torques and forces onto the body of the CM while it is being inserted into a constrained environment at the time of the collision with the environment. In addition, there should be information on where on the body of the CM, wrenches were applied. This may be accomplished, albeit with considerable difficulty in implementation, by knowing the exact model of the environment to detect the collocation places alongside the body of the CM in conjunction with observers or estimators to estimate forces and torques, or

another way would be to equip the CM with strain measurement sensors such as Fiber Bragg Grating (FBG) sensors in order to both detect the collocation spots and the amount of interaction forces.

A further crucial consideration is that the length of the CM being manipulated within a constraint workspace will vary, requiring the indices of discretized mesh elements to be modified accordingly. The scenario will be more challenging to deal with, particularly if the length of discretized elements is unequal.

The exploration of CMs in MIS and medical interventions presented in this work opens several avenues for future research. Reflecting on these findings, it becomes evident that there are key areas within CM modeling and control that require further investigation:

- Integration of Advanced Control Strategies: The advancements in control strategies for CMs, as discussed in earlier chapters, lay the groundwork for incorporating more sophisticated algorithms. The potential of machine learning and artificial intelligence, particularly reinforcement learning, emerges as a promising approach which could offer adaptive control capabilities in the dynamically challenging environments of human tissues, a topic that directly aligns with the control challenges highlighted in our experimental results.

- Improving Real-time Performance: Our research has underscored the criticality of real-time performance in CMs. There remains a gap in achieving the desired level of accuracy and responsiveness within the stringent constraints of surgical environments. Future work should pivot towards enhancing the computational efficiency of modeling and control algorithms and the real-time data processing capabilities of these systems, a need that was evident in our latency and response time analyses.

- Addressing the Simulation-Reality Gap: A notable observation from our experimental work is the divergence between simulation models and real-world application. This gap often results in performance discrepancies and unforeseen outcomes in surgical settings. Consequently, there is an imperative need for more robust simulation models and methodologies, aimed at bridging this gap and ensuring a smoother translation of simulated predictions to real-world surgical scenarios.

- Enhancing Autonomy in CMs: The potential of autonomous or semi-autonomous CMs to elevate the efficiency and safety of MIS has emerged as a critical discussion point. Future research directions could explore integrating sensory feedback, advanced imaging, and decision-making algorithms. Such integration would align with our findings on the limitations of manual control and the potential for enhanced automated responses.

- Material and Design Innovations: The development of new materials and design approaches for CMs could lead to significant advancements

in flexibility, durability, and bio-compatibility. This area of research is particularly promising, offering possibilities to overcome some of the mechanical limitations identified in our prototypes.

- Ethical and Regulatory Considerations: As the technology surrounding CMs progresses, there arises a concurrent need to address the ethical and regulatory dimensions. This includes considerations of patient safety, privacy, and the broader ethical implications of robotic interventions.
- Interdisciplinary Collaboration: The research conducted highlights the benefits of interdisciplinary collaboration, merging insights from robotics, medicine, material science, and computer science. Future endeavors in this field would benefit from continued and expanded collaborative efforts, fostering an environment where innovation can thrive.

In summary, these areas not only reflect the open issues and challenges encountered in our research but also pave the way for future explorations. They form the cornerstone for the next steps in CM research, particularly in the dynamic and evolving landscape of MIS.

In this chapter, we have journeyed through some discussions around the work conducted in this thesis, from the motivations driving the chosen modeling and control methods, to the performance and limitations of our research. As we conclude this chapter, it is clear that this thesis has contributed to the broader understanding and development of CM control. Despite the challenges and complexities faced, the potential for further exploration and improvement remains vast. This chapter has provided a context for the findings of our research, setting the stage for the final conclusions and future directions that will be presented in the next chapter.

References

[Bar08] Barrow, J. D. *New theories of everything: the quest for ultimate explanation*. 132. Oxford University Press, 2008.

[Box76] Box, G. E. “Science and statistics”. In: *Journal of the American Statistical Association* vol. 71, no. 356 (1976), pp. 791–799.

[Cox90] Cox, D. R. “Role of models in statistical analysis”. In: *Statistical Science* vol. 5, no. 2 (1990), pp. 169–174.

[Hag+10] Haga, Y. et al. “Active bending catheter and electric endoscope using shape memory alloy”. In: *Shape Memory Alloys*. IntechOpen, 2010.

[IIS02] Ikuta, K., Ichikawa, H., and Suzuki, K. “Safety-active catheter with multiple-segments driven by micro-hydraulic actuators”. In: *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2002: 5th International Conference Tokyo, Japan, September 25–28, 2002 Proceedings, Part I 5*. Springer. 2002, pp. 182–191.

[Kim+19] Kim, Y. et al. “Ferromagnetic soft continuum robots”. In: *Science Robotics* vol. 4, no. 33 (2019), eaax7329.

[LK04] Langelaar, M. and Keulen, F. van. “Modeling of a shape memory alloy active catheter”. In: *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*. 2004, p. 1653.

[NW72] Nelder, J. A. and Wedderburn, R. W. “Generalized linear models”. In: *Journal of the Royal Statistical Society: Series A (General)* vol. 135, no. 3 (1972), pp. 370–384.

[Ren+12] Renda, F. et al. “A 3D steady-state model of a tendon-driven continuum soft manipulator inspired by the octopus arm”. In: *Bioinspiration & biomimetics* vol. 7, no. 2 (2012), p. 025006.

[RWM18] Rich, S. I., Wood, R. J., and Majidi, C. “Untethered soft robotics”. In: *Nature Electronics* vol. 1, no. 2 (2018), pp. 102–112.

[Yun+12] Yun, C.-H. et al. “Multi-degree-of-freedom ultrasonic micromotor for guidewire and catheter navigation: The NeuroGlide actuator”. In: *Applied Physics Letters* vol. 100, no. 16 (2012), p. 164101.

Chapter 6

Concluding Remarks and Future Prospects

In this thesis, we scratched the surface of important challenges related to the modeling and control of magnetic catheters and guidewires, focusing on understanding the complex behavior of such tools. We explored topics across a series of studies, which hopefully can be used as inspiration in future academic research. This chapter focuses on summarizing our work, providing a connection between outcomes and knowledge of each individual study, and discussing them with respect to the research objectives formulated in the introduction (Chapter 1). Furthermore, to put our research in the relevant context, future prospects for where magnetically-actuated continuum manipulators are possible to develop in the near future are described.

6.1 Conclusions

Employing magnetic actuation opens up possibilities for the development of robotic catheters or guidewires with advanced functionalities but complex behavior. Controlled magnetic forces and torques can be applied locally and in a non-contact fashion to actuation points where magnetic elements such as permanent magnets or coils are located. This brings versatility to the magnetic actuation mechanism, which can be effectively used to move the CMs, change their formation, or make them interact with the environment directly. By addressing the research questions formulated in Section 1.4, we have intended to study the foundations for the modeling and control of CMs, upon which the complex behavior of magnetic CMs can be analyzed and put into effective and relevant clinical use. The key findings of the work presented to address the research questions of this dissertation are as follows:

To approach **RQ.1**, i.e., CM Modeling, we direct our focus towards the development of modeling solutions to capture the mechanics of CMs using the principles of continuum robotics. In Chapter 3 and its related papers Paper I, Paper II, Paper IV, and Paper V, we initiate our work mainly on developing nonlinear dynamic models that are Lie group variational and recurrent neural network integrations. A significant amount of work within this chapter is devoted to formulating an exact model of CMs. We presented the derivation and experimental validation of algorithms for modeling and estimation of magnetically-actuated soft continuum manipulators using Lie group variational integration. Using magnetic actuation, dynamic and static experiments were conducted on manipulators with rigid and soft materials (e.g., aluminum and PDMS) to illustrate the validity of the presented algorithm for a wide range of

experiments.We identified several principal contributions to our work. First of all, we derived and validated a fully three-dimensional dynamic model for forced continuum manipulators with soft materials undergoing spatial deformation. Secondly, we proposed distributed predictive filters to capture the behavior of internal and external dissipative damping. Finally, we validated our approach by designing and carrying out different experiments with metal rods and polymer-based soft rods.

In Section 3.2, the proposed exact model is used to realize a real-time model for forced CMs undergoing spatial deformation, making the model usable in closed-loop control applications. In data-driven approaches such as recurrent neural networks, there is a need for a large number of labeled data. This is usually a barrier to using neural networks in robotic applications. We used the Lie group variational integration method, presented earlier, to generate the necessary data-set for training recurrent neural network-based models. It was also shown that the developed models could be generalized and applied to different boundary and initial conditions without the need for re-training. Furthermore, the computation time to run these neural-network-based models is much faster (and in some examples, a few orders of magnitude less) than the time needed to run the exact nonlinear Lie group-based method. Such a large computation bandwidth is beneficial for employing such models in closed-loop control tasks. We finally demonstrate that the presented model can outperform classical modeling approaches such as the Cosserat rod model while also showing possibilities for the model being used in practice by providing high quality predictions of CMs' behavior[1].

Permanent magnets or electromagnetic coils located outside of the body are auxiliary infrastructures that are used to generate external fields penetrating through the body and transmitting power to manipulate CMs at their actuation points (which can either be small coils or permanent magnets integrated within the bodies of CMs). In the third part, we explore a rapid multiphysics model by employing the dipole and linear elasticity theories to capture the mechanics of magnets (auxiliary infrastructures) and guidewires, respectively. This provides a quick tool to roughly simulate both magnetically actuated CMs and the necessary auxiliary infrastructure for the manipulation of CMs.

Within the boundaries of the research question **RQ.2**, i.e., CM Control, we explore control and path planning strategies for manipulation of untethered smallscale robots, which essentially can be seen as a magnetic actuation point for CMs. Such path planners eventually will contribute to making magnetic CMs operational in a (semi-)autonomous mode. In Chapter 4 and its related papers, Paper III and Paper V, three path planning strategies are developed and compared. However, the main effort involves deriving and experimenting with a deep reinforcement learning method. We utilize a customized Rainbow algorithm, along with Quantile Huber loss and ResNet, to bridge the gap between simulations and real-world implementation of a reinforcement learning algorithm for a robotic application. This application focuses on particle deliveries

[1]See Table II.1 and Figure II.14 for a detailed performance comparison.

within a simplified model of the large intestine, which represents a constrained and uncertain environment. The experimental results on the UR5 robot show an average success rate of 98.86% over 30 episodes for randomly generated trajectories, demonstrating the viability of the proposed approach for real-life applications. To evaluate the results of the RL-based mechanism, two classical approaches, Attractor Dynamics and Execution Extended Rapidly-Exploring Random Trees (ERRT), are tested through simulations and experiments. The results confirm that the RL-based method achieves comparable performance while being a more robust and generic solution to deploy in dynamic and complex environments. Eventually, by using the proposed rapid kinematic model, an optimal angle for orienting an external magnetic field is calculated, which results in a desired shape of continuum guidewire for being steered into a vascular-like structure.

6.2 Future Work

The work done in this dissertation brings some insights into how CMs can be analyzed by investigating modeling and control strategies. The subject of CMs, especially magnetically actuated CMs, is evolving in many areas, particularly in the field of minimally invasive surgeries. According to our experience, some educated potential suggestions are provided to promote that evolution.

The complex mechanical behavior of magnetic CMs, as it is discussed throughout the chapters, raises expectations that these instruments can perform a wide variety of clinically-relevant tasks in challenging areas within the human body. Yet, related challenges must be tackled to be able to simulate and analyze the dynamics and behavior of such tools. Although the presented analytical model in this work tries to model the mechanics of CMs exactly, there will be challenges, such as reducing computation bandwidth while capturing the exact dynamics of CMs and preserving their outstanding conservation properties. This can be done by employing asynchronous variational integrators in a distributed, parallel scheme for computations.

It was demonstrated that in the modeling of CMs —discretized as several nodes—, incorporating poses of adjacent nodes and also wrenches as separated inputs helps to have a real-time generalizable model rather than just purely learning the structure of data, while achieving high quality predictions. However, supervised learning methods likely tend to preserve the structure of data, and these models might not entirely respect the underlying physics (conservation laws). In other words, these methods might not be wholly physics-aware and applicable for untrained or unprecedented dynamics or geometries without any adjustment, re-training, or using techniques such as transfer learning. One possible solution that can be considered to tackle the mentioned issue and move toward fully physics-aware neural networks is revisiting lost functions for the training process to enforce the structure imposed by PDE equations describing CMs dynamics and kinematics such that the trained neural network will be aware of governing PDEs.

A plausible path to bio-active medical robotic agents goes through smallscale robot technology aiming at utilizing micro- and nano-robots to reach sites of treatments in the human body. As we demonstrated in Chapter 4, magnetic particles can serve as controllable agents in clinically-related tasks. It is worth mentioning that flexible continuum tools and smallscale robots can be seen as inseparable elements of the same bio-active medical tool. Furthermore, investigating suitable control and path-finding strategies for such a system is therefore a necessity.

Although deep reinforcement learning methods are widely applied in simulations, employing these algorithms for robotic applications is challenging, both due to the difficulty of obtaining training samples and the lack of a concrete analytical proof for ensuring the stability of a system. Yet, the convergence of the system to a stable behavior can be shown statistically to a very good extent. Furthermore, there is no need for a precise model for the robot and its interaction with the environment, which can make reinforcement learning-based methods a general solution for tackling challenges in controlling robots where classical control might be difficult to apply. This comes at a price that there should be large enough data-sets to train robots on, which would be costly in real-world robotic applications. More improvements are possible, both with respect to the techniques used in the design of reinforcement learning-based algorithms and to making tasks and environments more realistic for robots. For example, three-dimensional dynamic environments or obstacles that change over time can be investigated. Equipping the RL method with a high fidelity segmentation algorithm would help in better understanding the environment, thereby reducing the simulation reality gap. In addition, for more practical uses, all the imaging modalities, i.e., tracking systems, are better integrated at the end-effector and replaced by more relevant radiography methods, which are considered safe procedures and can be used to see through the body. It was observed that offline training of high fidelity reinforcement learning algorithms in complex environments requires a relatively large amount of time. The training time can be reduced by augmenting a model or pre-existing knowledge, such as a classical path-finding method such as the presented ERRT with learning methods.

Papers

Paper I

Dynamic Modeling of Soft Continuum Manipulators Using Lie Group Variational Integration

Abbas Tariverdi[1], **Venkatasubramanian K. Venkiteswaran, Ørjan G. Martinsen, Ole J. Elle, Jim Tørresen, Sarthak Misra**

Published in *PLOS ONE*, June 2020, volume 15, issue 7, pp. e0236121. DOI: 10.1371/journal.pone.0236121.

Abstract

This paper presents the derivation and experimental validation of algorithms for modeling and estimation of soft continuum manipulators using Lie group variational integration. Existing approaches are generally limited to static and quasi-static analyses, and are not sufficiently validated for dynamic motion. However, in several applications, models need to consider the dynamical behavior of the continuum manipulators. The proposed modeling and estimation formulation is obtained from a discrete variational principle, and therefore grants outstanding conservation properties to the continuum mechanical model. The main contribution of this article is the experimental validation of the dynamic model of soft continuum manipulators, including external torques and forces (e.g., generated by magnetic fields, friction, and the gravity), by carrying out different experiments with metal rods and polymer-based soft rods. To consider dissipative forces in the validation process, distributed estimation filters are proposed. The experimental and numerical tests also illustrate the algorithm's performance on a magnetically-actuated soft continuum manipulator. The model demonstrates good agreement with dynamic experiments in estimating the tip position of a Polydimethylsiloxane (PDMS) rod. The experimental results show an average absolute error and maximum error in tip position estimation of 0.13 mm and 0.58 mm, respectively, for a manipulator length of 60.55 mm.

The paper was partially supported by the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation programme (Grant Agreement #638428—project ROBOTAR).

[1]University of Oslo, Postboks 1337 Blindern, 0316 Oslo, Norway, abbast@uio.no

Contents

I.1 Introduction 80
I.2 Continuum Manipulator Dynamics 83
I.3 Lie Group Variational Integrators for the Forced Continuum Manipulator 86
I.4 Simulation and Experimental Results 94
I.5 Discussion 104
I.6 Conclusions and Future Work 110
I.A Preliminaries on Lie groups and Lie algebra 111
References 111

I.1 Introduction

Reachability, high level of dexterity, and large elastic deformability are the primary driving factors behind the growth of research in the design, modeling, and control of continuum manipulators. Flexible continuum manipulators have recently generated interest in several fields [LMC16; Pol+17; RT15], especially in minimally invasive surgical robotics and interventional medicine, such as catheter-based endovascular intervention [Bur+13; Gra+00] and cardiac surgeries [CP09; KH11]. In contrast to conventional rigid link manipulators, soft manipulators are able to reshape their configurations to allow for redundancies in path planning, and are capable of precise and delicate manipulation of objects in complex and varying environments.

There are numerous candidate actuation mechanisms for continuum manipulators such as tendon-drives and concentric tubes [Cam+08; Cia+13; Guo+96; WJ10]. Compared to other actuation mechanisms, magnetic actuation benefits from high dexterity, versatility, and wireless actuation [HSM18; Sik+17; Sik+19; Tho+20]. By applying remote magnetic torques on permanent magnets or coils which are embedded inside the body of a manipulator and/or at its tip, one can control the motion and configuration of the manipulator.

This paper aims to develop a computational model for analyzing the dynamics of soft continuum manipulators, which is one of the key challenges in soft robotics. In many tasks, dynamic models of manipulators are essential for control, trajectory planning, and optimal design purposes, especially in Minimally Invasive Surgeries (MIS) for operation in unknown and unstructured environments such as inside the human body. Due to elastic characteristics and geometric nonlinearities (i.e., bending, torsion, shear, elongation, including large deformation) of continuum manipulators, their dynamics have highly nonlinear behavior and are expressed as partial differential equations. Some recent modeling approaches of soft continuum manipulators/robots, which have been employed in the surgical robotics field, are summarized in Table I.1.

Table I.1: References on dynamics/static analysis of soft continuum manipulators in surgical robotics field.

References	**Modeling Approach and Properties**	**Robot Type/Application**
[Pai02]	Static analysis: Cosserat rod model. 3D elesticity	Surgical suture/ strands
[WRC09]	Beam mechanics based on elastic energy	Concentric tubes/ General MIS
[XS08]	Static analysis based on screw theory and a virtual-work model	Multiple parallel backbones/ General MIS
[Cam+08; CCS09]	Linear elasticity theory	Single/Redundant tendons
[JGT09]	3D Static analysis with loads: Cosserat rod model	General purpose CRs
[Ruc+10]	Beam mechanics based on elastic energy (includes both bending and torsion)	Concentric tubes/ General MIS
[XS10]	Bernoulli–Euler elastica theory: statics, 2D	Multibackbone
[KS10]	Static analysis based on a virtual-work model	Serial Segments/ General surgical end-effectors
[Loc+10; RJW10]	Static analysis: Cosserat rod theory	Concentric tubes with and without external loads
[Tun11]	Static analysis: Cosserat rod theory	Magnetic Catheter/ General purposes
[MD11]	Loaded static analysis: Cosserat rod theory	General MIS
[Wen+12]	Dynamic analysis: Cosserat rod model. 3D elesticity	Guidewire/ Interventional Radiology procedures
[Tun13]	FEM: large deformation and inflation	Simulations on general medical robots
[JPZ14]	Lumped-parameter model	Multiple parallel shafts/ general Magnetic resonance imaging (MRI)-compatible medical manipulators
[GC15]	Pseudo-rigid-body model	Multiple parallel shafts/ cardiac robotic catheter
[RM16]	2D static analysis: rigid-link modeling	Planar tendon-driven continuum manipulator/ general medical robots
[GDS19]	Static analysis: α Lie group formulation	Planar continuum: simulations and bechmark analysis/ intravascular shaping operations
[VSM19]	3D static analysis: pseudo rigid body model	Magnetic catheter/ General surgical catheters

Soft continuum manipulators are analogous to specific Cosserat continuums. Therefore, Lie group synchronous variational integrators [BS99; Dem+15], a novel time and space integration scheme, is employed in this paper to model geometrically exact beams based on the Simo beam model [Sim85] and Hamiltonian formulation. The core idea of this algorithm is to obtain the dynamic behavior of the system while conserving the invariants (energy, momentum maps) of the system, especially for long-time simulations. The distinguishing characteristic of variational integrators is that they define the equations of motion based on the discretized variational principle of the system. Combining the integrators with Lie-group/algebraic techniques enables the algorithm to respect not only the structure of the dynamics and its properties but also preserve the structure of the configuration space. The advantages of employing the Lie group variational integration method compared to other modeling strategies is that the proposed solver is applicable to exact nonlinear dynamic models of continuum manipulators subject to large deformations. The algorithm preserves the symplectic structure, i.e., the invariants of mechanical systems. Also, it allows the usage of different time steps at different points in a given finite element for the geometry of soft manipulators. These properties are investigated in previous work (e.g., [Dem+15; Lee08; LOL14]), while the main focus of this paper is the experimental validation of the method on magnetically-actuated soft continuum manipulators.

Investigation of previous work in modeling of the continuum manipulators suggests that existing literature focuses primarily on static or quasi-static approaches, or does not provide sufficient experimental validation in realistic application scenarios. By contrast, the main contribution of this article compared with the existing work in literature is the validation of an accurate dynamic model of a soft continuum manipulator, considering spatial motion. Also, it should be noted that the model accounts for the geometric nonlinearities (e.g., large deformation) and respects conservation of dynamical properties of the system (e.g., energy and momentum maps conservation), and structures of configuration space simultaneously. Besides, it should be pointed out that three-dimensional internal and external dissipation forces act on the continuum manipulator and hence affect the dynamics. Therefore, it is necessary to consider these friction/dissipation forces in the validation process. To this end, distributed prediction filters have been proposed.

In summary, this article's contributions can be stated as follows.

- Existing studies on the modeling of continuum manipulators primarily consider static or quasi-static approaches. However, in numerous applications, the fully spatial dynamics of manipulators need to be considered for accurate control and design purposes. The primary contribution of this article is the derivation and experimental validation of a dynamic model for forced continuum manipulators with soft materials undergoing spatial deformation. The model accounts for the nonlinearities of the continuum manipulator; loading resulted from magnetic fields, the gravity, and internal and external dissipation forces generated by friction.

- Due to the difficulty in obtaining knowledge about the internal and external dissipation forces, distributed estimation filters have been designed to take these forces into account and capture their behavior.

The rest of the paper is organized as follows. In Section I.2, mathematical preliminaries, including the system description and notation, are discussed. Next, Section I.3 addresses the algorithm and numerical results. The experimental framework and implementation results are described in Section I.4, demonstrating the effectiveness of the theoretical formulation. In addition, Section I.5 provides a discussion on the implementation of the modeling algorithm. Finally, Section I.6 summarizes the results of this work and draws conclusions and posits directions for future work.

I.2 Continuum Manipulator Dynamics

This section is devoted to describing kinematics and full three-dimensional dynamics for continuum manipulators undergoing large deflections (for detailed explanations, refer to the reference [Dem+15]). We review the static description of a continuum manipulator in three-dimensional space $\mathbb{R}^3$ toward deriving the dynamic equations of motion of geometrically exact continuum manipulator by applying Hamilton's principle to the Lagrangian of the system.

I.2.1 Kinematics

The manifold of configuration space of a continuum manipulator considering Boundary Conditions (BCs) is defined as

$$\mathcal{Q} = \Big\{ (\mathcal{O}, \mathcal{P}) \in \mathcal{C}^{\infty}(\cdot) : [0, L] \to \mathcal{SO}(3) \times \mathbb{R}^3 \Big| \textit{BCs are satisfied} \Big\}$$

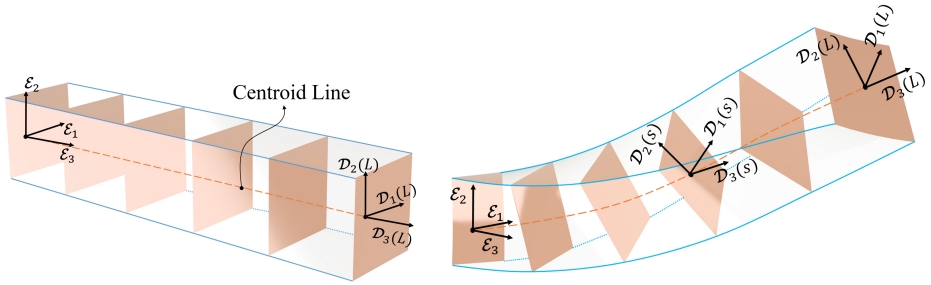
in which L is the length of the undeformed continuum manipulator, $\mathcal{P} : [0, L] \to \mathbb{R}^3$ maps the line of continuum manipulator's centroids (i.e. center of mass) to Euclidean space $\mathbb{R}^3$ and the orthogonal transformation $\mathcal{O} : [0, L] \to \mathcal{SO}(3)$ determines the orientation of moving cross-sections at points $\mathcal{P}(s)$ in the terms of a fixed basis $\{\mathcal{E}_1(s), \mathcal{E}_2(s), \mathcal{E}_3(s)\}$. Therefore, the orientation of each cross-section which is denoted by directors or moving basis $\{\mathcal{D}_1(s), \mathcal{D}_2(s), \mathcal{D}_3(s)\}$ can be written as

$$\mathcal{D}_i(s) = \mathcal{O}(s)\mathcal{E}_i, \ i = 1, 2, 3.$$

Figure I.1 shows initial and a time-evolved configuration of the continuum manipulator with the free right tip and clamped left end, i.e, BCs: $\mathcal{O}(0) = I_3$, $\frac{\partial \mathcal{P}(0)}{\partial s} = \mathcal{E}_3$. The BCs imply that the clamped cross section is orthogonal to the plane defined by $\mathcal{E}_1$ and $\mathcal{E}_2$. In addition, a curve $q(s, t) = (\mathcal{O}(s, t), \mathcal{P}(s, t)) \in \mathcal{Q}$ characterizes a time-evolved configuration space of the continuum manipulator. The family of tangent vectors to the curve $q(t)$ is defined as

$$\dot{q}(s, t) = \frac{\mathrm{d}q(s, t)}{\mathrm{d}t} = \Big(\dot{\mathcal{O}}(s, t), \dot{\mathcal{P}}(s, t) \Big) \in T_q\mathcal{Q},$$

which characterize tangent bundle $T_q\mathcal{Q}$ to $\mathcal{Q}$ at the manifold $q(s, t)$.



(a) Initial Configuration (b) Time-Evolved Configuration

Figure I.1: Initial and time-evolved configurations of the continuum manipulator. The highlighted frames depict cross-sections at discretization points. Fixed bases or material frame $\{\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3\}$ are also shown at the fixed end of the manipulators. Also, moving bases $\{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3\}$ are attached to the cross section at the centroid s and the tip of manipulators.

I.2.2 Lagrangian and Equation of Motion

To derive the equations of motion, we first need to introduce the Lagrangian $\mathcal{L} : T_q\mathcal{Q}$ of the system which can be written as

$$\mathcal{L}(\mathcal{O}, \mathcal{P}, \dot{\mathcal{O}}, \dot{\mathcal{P}}) = \underbrace{\frac{1}{2}\int_0^L \Big(M\|\dot{\mathcal{P}}\|^2 + \omega^T J\omega\Big)\mathrm{d}s}_{\text{Kinetic energy}} - \underbrace{\frac{1}{2}\int_0^L \Big((\Gamma - \mathcal{E}_3)^T C_1(\Gamma - \mathcal{E}_3) + \Omega^T C_2\Omega\Big)\mathrm{d}s}_{\text{Elastic energy}} - \underbrace{\int_0^L F^c \cdot \mathcal{P}\mathrm{d}s}_{\substack{\text{Conservative} \\ \text{potential energy}}} \quad \text{(I.1)}$$

where the matrices C_1 and C_2 are defined as $C_1 := \mathrm{diag}(GA\ GA\ EA)$ and $C_2 := \mathrm{diag}(EI_1\ EI_2\ GJ^p)$. For brevity, other parameters are defined in Table I.2.

Table I.2: Definition of parameters in Lagrangian (Equation (I.1)) as described in [Dem+15].

$M = \rho_0 \times A$	ρ_0 and A are the body constant mass density and cross section's area.
$\omega(s,t) \in \mathfrak{so}(3)$	the body angular velocity
$J = -\rho_0 \int_{\mathcal{A}} \widehat{(x\mathcal{E}_1 + y\mathcal{E}_2)}^2 dxdy$	inertia matrix in the fixed frame
$(\Omega(s,t), \Gamma(s,t)) = (\mathcal{O}^{-1}\frac{\partial \mathcal{O}}{\partial s}, \mathcal{O}^{-1}\frac{\partial \mathcal{P}}{\partial s})$	deformation gradients as viewed at the time t by an observer that is located at the position s
E, $G = E/(2(1+\nu))$, ν, I_1, I_2, and J^p	Young's modulus, shear modulus, Poisson's ratio, principal moments of inertia of the cross-section, and polar moment of inertia, respectively.

where each cross section is given by a compact set $\mathcal{A} = \{(x,y)|x,y \in \mathbb{R}\}$, Lie algebra $\mathfrak{so}(3)$ is associated with the Lie group $SO(3)$, and Hat map/ operator $\hat{} : \mathbb{R}^3 \to \mathfrak{so}(3)$ which is a one-to-one invertible map, i.e., an isomorphism, is defined as

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \to \hat{\mathbf{v}} = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \tag{I.2}$$

.

Remark I.2.1. the inertia J dependents both on the cross-sectional area A and also on the distribution of mass within the cross-section. The integral takes into account the coordinates x and y within the cross-sectional area A, which corresponds to how the mass is distributed in relation to the $\mathcal{E}_1$ and $\mathcal{E}_2$ axes. The inertia matrix J takes into consideration the geometry of the cross-section, including how far from the center of mass (centroid) the material is distributed, as this affects the body's resistance to angular acceleration. The use of the hat operator $\hat{}$ implies that it is considering the rotational effects due to the distribution of mass around each axis, not just the area alone.

For the continuum manipulator described, which is naturally aligned with $\mathcal{E}_3$, and has cross-sections at discretization points where $\mathcal{E}_3$ is normal to the cross-sections, Inertia is most significant about the principal axes of the cross-section, which are typically aligned with the manipulator's geometric axes. For a continuum manipulator, the principal moments of inertia are often most relevant in the plane of the cross-sections (i.e., $\mathcal{E}_1 - \mathcal{E}_2$ plane), where each discretized node is expected to experience the most significant bending deformations. Each discretized node, considering its cross-section, is designed to bend primarily in the $\mathcal{E}_1 - \mathcal{E}_2$ plane, which means that inertia about the $\mathcal{E}_3$ axis (normal to this plane) is most relevant for the dynamics of bending motions for each node.

The Euler-Lagrange equations are obtained by applying by the Lagrange-d'Alembert principle to the action functional $\mathcal{H}$ associated to $\mathcal{L}$, namely

$$\mathcal{Y}(\mathcal{O},\mathcal{P}) = \int_{t_o}^{t_f} \Big(\mathcal{L}(\mathcal{O},\mathcal{P},\dot{\mathcal{O}},\dot{\mathcal{P}}) + F^{nc}(\mathcal{O},\mathcal{P},\dot{\mathcal{O}},\dot{\mathcal{P}})\Big)\mathrm{d}t$$

By employing the Lagrange-d'Alembert principle, one computes

$$\begin{aligned}\delta\mathcal{Y} = \int_{t_o}^{t_f} \Bigg(& \int_0^L \Big(\mathcal{M}\dot{\mathcal{P}}^T(\delta\dot{\mathcal{P}}) + \omega^T J\delta\omega\Big)\mathrm{d}s \\ & - \int_0^L \Big((\Gamma-\mathcal{E}_3)^T C_1\delta\Gamma + \Omega^T C_2\delta\Omega\Big)\mathrm{d}s \\ & - \int_0^L F^c\delta\mathcal{P}\ \mathrm{d}s - F^{nc}\cdot\delta q(s,t)\Bigg)\mathrm{d}t\end{aligned}$$

The terms $\delta\omega$, $\delta\Omega$, and $\delta\Gamma$ are defined ([LOL14]) as follows:

$$\begin{aligned}\delta\omega &= \omega\times\eta + \frac{\mathrm{d}}{\mathrm{d}t}\eta \\ \delta\Omega &= \frac{\partial}{\partial s}\eta + \Omega\times\eta \\ \delta\Gamma &= \mathcal{O}^T\delta\Big(\frac{\partial}{\partial s}\mathcal{P}\Big) + \Gamma\times\eta\end{aligned} \tag{I.3}$$

where $\delta\mathcal{O} = \mathcal{O}\hat{\eta}$.

Taking into account the expressions for $\delta\omega$, $\delta\Omega$, and $\delta\Gamma$ in Equations (I.3) and using integration by parts in space and time, we obtain Euler-Lagrange equations with non-conservative force $F^{nc}(\mathcal{O},\mathcal{P},\dot{\mathcal{O}},\dot{\mathcal{P}}) : T_q\mathcal{Q}\to T_q^*\mathcal{Q}$ as

$$\begin{aligned}J\dot{\omega} - J\omega\times\omega - \Gamma\times C_1(\Gamma-\mathcal{E}_3) - \Omega\times C_2\Omega - C_2\frac{\partial\Omega}{\partial s} &= \mathcal{O}^{-1}\mathcal{N} \\ M\ddot{\mathcal{P}} - \frac{\partial\mathcal{O}C_1(\Gamma-\mathcal{E}_3)}{\partial s} + F^c &= \mathcal{F}\end{aligned} \tag{I.4}$$

in which we could consider 6×1 representations of general non-conservative force vector $F^{nc} = \left[\begin{smallmatrix}\mathcal{N}\\ \mathcal{F}\end{smallmatrix}\right]$ where $\mathcal{F}$ and $\mathcal{N}$ are force and moment vectors in $\mathbb{R}^3$, respectively. Also, $T_q^*\mathcal{Q}$ denotes the cotangent bundle of $\mathcal{Q}$. For simplicity, one may think of the cotangent boundle as the space of positions and momenta. For the exact definitions, refer to [MR94] or [AM78].

I.3 Lie Group Variational Integrators for the Forced Continuum Manipulator

In this section, the focus is on analyzing a Lie group variational integrator for continuum manipulators with conservative (e.g., the gravity) and non-conservative forces (e.g., friction and loads inserted by actuators). In the following

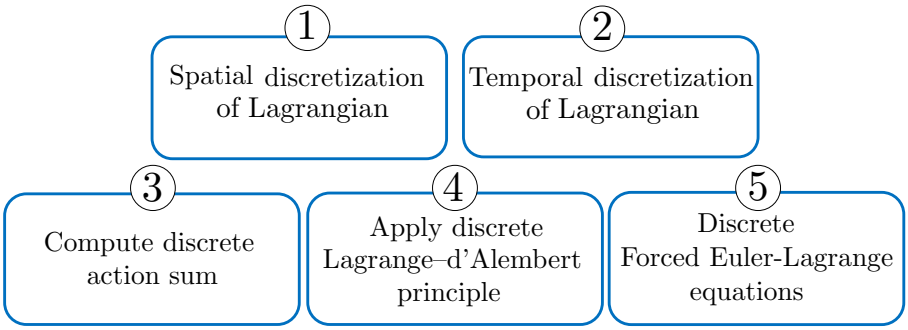


Figure I.2: Steps 1 through 5 toward deriving continuum manipulator discrete dynamics.

subsection, the discretized version of the forced Euler-Lagrange Equation (I.4) is given (for further details and stability analysis, see [Dem+15]), and afterward, the estimation process is discussed.

I.3.1 Modeling

This section is devoted to introducing a Lie group variational integration scheme for continuum manipulators with external loading. First, one needs to consider the spatial discretization of Lagrangian introduced in the previous section. Afterward, discrete Lagrange-d'Alembert equations need to be expressed on Lie group $SE(3)$. These equations are employed to propose a model-based distributed estimation scheme. Figure I.2 depicts the modeling procedure in this section.

Here notations of the paper are provided. Additionally, concepts and definitions on Lie groups and Lie algebra are presented in Appendix I.A.

Notations: The undeformed continuum manipulator's length $[0, L]$ is spatially discretized into N subsets $\mathcal{I}_i = [s_{a_i}, s_{a_{i+1}}]$ of length $l_{\mathcal{I}_i} = s_{a_{i+1}} - s_{a_i}$. For an element $\mathcal{I}_i$, a_i and a_{i+1} denote its left and right nodes. The configuration of the continuum manipulator at the node a_i is given by $\mathcal{O}_{a_i} := \mathcal{O}(s_{a_i})$ and $p_{a_i} := \mathcal{P}(s_{a_i})$. Also, ω_{a_i} denotes the angular velocity of a node a_i. Given a node a_i, the discrete time evolution of this node is given by the discrete curve $(\mathcal{O}^j_{a_i}, p^j_{a_i}) \in SE(3) = SO(3) \times \mathbb{R}^3,\ j = 0, \cdots, V$ and is based on the discrete Euler-Lagrange equations on Lie group $SE(3)$, The discrete variables $F^j_{a_i}$ associated with the node a_i are defined as $F^j_{a_i} = (\mathcal{O}^j_{a_i})^T \mathcal{O}^{j+1}_{a_i}$. We denote the fixed time step by $\Delta t = t_j - t_{j+1},\ j = 0, \cdots, V$. In time discretization of the continuum manipulator, we have $\Delta p^j_{a_i} := p^{j+1}_{a_i} - p^j_{a_i}$.

By identifying the configuration space $\mathcal{Q}$ of the continuum manipulator with the infinite dimensional Lie group $G = \mathcal{C}^\infty([0, L], \mathcal{SO}(3) \times \mathbb{R}^3$, we consider the trivialized Lagrangian $\mathscr{L} : G \times \mathfrak{g} \to \mathbb{R}$, where $\mathfrak{g}$ is a Lie algebra associated

with the Lie group G. A spatial discretization of the trivialized Lagrangian for an element $\mathcal{I}_i$ and the total system are computed as follows, respectfully. It should be noted that the evaluation of Lagrangian at midpoints of nodes is employed. Other evaluations of the Lagrangian depending on a different number or combinations of nodes are possible (see [LOL14]).

For an element $\mathcal{I}_i$:

$$\mathscr{L}_{\mathcal{I}_i} = \frac{l_{\mathcal{I}_i}}{4} M\Big(\|p_{a_i}\|^2 + \|p_{a_{i+1}}\|^2\Big) + \frac{l_{\mathcal{I}_i}}{4}\Big(\omega_{a_i}^T J \omega_{a_i} + \omega_{a_{i+1}}^T J \omega_{a_{i+1}}\Big) - \mathcal{V}_{\mathcal{I}_i} \quad \text{(I.5)}$$

where $\mathcal{V}_{\mathcal{I}_i}$ is conservative potential energy of an element $\mathcal{I}_i$ due to the gravity and elasticity and given by

$$\mathcal{V}_{\mathcal{I}_i} = \frac{l_{\mathcal{I}_i}}{4}\left[\Big(\mathcal{O}_{a_i}^T \frac{\Delta p_a}{l_{\mathcal{I}_i}} - \mathcal{E}_3\Big)^T C_1 \Big(\mathcal{O}_{a_i}^T \frac{\Delta p_a}{l_{\mathcal{I}_i}} - \mathcal{E}_3\Big) + \Big(\mathcal{O}_{a_{i+1}}^T \frac{\Delta p_a}{l_{\mathcal{I}_i}} - \mathcal{E}_3\Big)^T C_1 \Big(\mathcal{O}_{a_{i+1}}^T \frac{\Delta p_a}{l_{\mathcal{I}_i}} - \mathcal{E}_3\Big)\right] \quad \text{(I.6)}$$

For the whole continuum manipulator:

$$\mathscr{L} = \sum\nolimits_{i=1}^{N}\Big(\frac{l_{\mathcal{I}_i}}{2} M\|p_{a_i}\|^2 + \frac{l_{\mathcal{I}_i}}{2}\omega_{a_i} J \omega_{a_i}\Big) + \sum\nolimits_{i=0,N+1}\Big(\frac{l_{\mathcal{I}_i}}{4} M\|p_{a_i}\|^2 + \frac{l_{\mathcal{I}_i}}{4}\omega_{a_i} J \omega_{a_i}\Big) - \sum\nolimits_{i=0}^{N+1} \mathcal{V}_{\mathcal{I}_i}$$

The temporal discretized Lagrangian $\mathscr{L}_{\mathcal{I}_i}^j$ approximates the Lagrangian $\mathscr{L}_{\mathcal{I}_i}$ in Equation (I.5) during the time step Δt is therefore

$$\mathscr{L}_{\mathcal{I}_i}^j = \sum_{a=a_i,a_{i+1}}\Big(\frac{l_{\mathcal{I}_i}}{4} M \frac{\|H_a^j\|^2}{\Delta t} + \Big(\frac{l_{\mathcal{I}_i}}{2}\frac{\mathrm{Trace}\big((I_3 - F_a^j)J_d\big)}{\Delta t}\Big) - \Delta t \mathcal{V}_{\mathcal{I}_i}^j \quad \text{(I.7)}$$

where $H_a^j = (\mathcal{O}_a^j)^T \Delta p_a^j$ and $J_d = \frac{\mathrm{Trace}(J)}{2} I_3 - J$.

The discrete action sum over the discretized time interval $[0, T] = \{t^0, \cdots, t^j | t^j = t^{j-1} + \Delta t, t^0 = 0, t^V = T\}$, is computed as follows.

$$\mathcal{Y}_d = \sum_{i=0}^{N+1}\sum_{j=1}^{V} \mathscr{L}_{\mathcal{I}_i}^j$$

The discrete Lagrange–d'Alembert principle is

$$\delta \sum_{i=0}^{N+1}\sum_{j=0}^{V} \mathscr{L}^j + \sum_{j=0}^{V}\sum_{i=0}^{N+1} F_{d a_i}^{ncj} \cdot \delta(\mathcal{O}_{a_i}^j, p_{a_i}^j) = 0 \quad \text{(I.8)}$$

By applying the discrete Lagrange–d'Alembert principle (I.8), we get the discrete Euler–Lagrange equations for a node a_i in a compact form as

$$
\begin{aligned}
&T_e^* L_{(F_{a_i}^{j-1},H_{a_i}^{j-1})}\Big(D_{F_{a_i}^{j-1}}\mathscr{L}_{a_i}^{j-1}, D_{H_{a_i}^{j-1}}\mathscr{L}_{a_i}^{j-1}\Big)\\
&- Ad^*_{(F_{a_i}^{j},H_{a_i}^{j})^{-1}} T_e^* L_{(F_{a_i}^{j},H_{a_i}^{j})}\Big(D_{F_{a_i}^{j}}\mathscr{L}_{a_i}^{j}, D_{H_{a_i}^{j}}\mathscr{L}_{a_i}^{j}\Big)\\
&+ T_e^* L_{(\mathcal{O}_{a_i}^{j},p_{a_i}^{j})}\Big(D_{\mathcal{O}_{a_i}^{j}}\mathscr{L}_{a_i}^{j}, D_{p_{a_i}^{j}}\mathscr{L}_{a_i}^{j}\Big) + (\mathcal{O}_{a_i}^{j}, p_{a_i}^{j})^{-1} F_{d\,a_i}^{ncj} = 0
\end{aligned}
\tag{I.9}
$$

Finally, using the definitions of adjoint and coadjoint actions, and cotangent lift of left translation which are presented in Appendix I.A, Equations (I.7) and (I.9) yields Equations (I.10)-(I.12) and (I.14)-(I.16) to update rotations and positions of each node.

I.3.1.1 Discrete Euler-Lagrange Equations for Rotations

- For the left node of the continuum manipulator $(a_{i=0})$

$$
\begin{aligned}
(F_{a_0}^j J_d - J_d (F_{a_0}^j)^T)^\vee = -\frac{2\Delta t^2}{l_{\mathcal{I}_0}}\Bigg[&\frac{1}{2}C_1\Big(\mathcal{O}_{a_0}^T\frac{\Delta p_{a_0}}{l_{\mathcal{I}_0}} - \mathcal{E}_3\Big)\times \mathcal{O}_{a_0}^T\Delta p_{a_0}\\
&+\frac{1}{l_{\mathcal{I}_0}}\Big(\big((I+\mathcal{O}_{a_{0+1}}^T\mathcal{O}_{a_0})^{-1}\widehat{C_2\psi_{a_0}}(\hat{\psi}_{a_0}-2I)\big)^{(A)}\Big)^\vee\\
&-\Delta t\mathcal{O}_{a_0}^{-1}\mathcal{N}_{a_0}\Bigg]\Bigg|_{t=t^j} + (J_d F_{a_0}^{j-1} - (F_{a_0}^{j-1})^T J_d)^\vee
\end{aligned}
\tag{I.10}
$$

- For the interior nodes of the continuum manipulator $\forall a_i,\ i\in\{1,\cdots,N-1\}$

$$
\begin{aligned}
(F_{a_i}^j J_d - J_d (F_{a_i}^j)^T)^\vee = -\frac{\Delta t^2}{l_{\mathcal{I}_i}}\Bigg[&\frac{1}{2}C_1\Big(\mathcal{O}_{a_i}^T\frac{\Delta p_{a_{i-1}}}{l_{\mathcal{I}_i}} - \mathcal{E}_3\Big)\times \mathcal{O}_{a_i}^T\Delta p_{a_{i-1}}\\
+\frac{1}{2}C_1\Big(\mathcal{O}_{a_i}^T\frac{\Delta p_{a_i}}{l_{\mathcal{I}_i}} - \mathcal{E}_3\Big)\times \mathcal{O}_{a_i}^T\Delta p_{a_i} &+ \frac{1}{l_{\mathcal{I}_i}}\Big(\big((I+\mathcal{O}_{a_{i+1}}^T\mathcal{O}_{a_i})^{-1}\widehat{C_2\psi_{a_i}}(\hat{\psi}_{a_i}-2I)\big)^{(A)}\Big)^\vee\\
+\frac{1}{l_{\mathcal{I}_i}}\Big(\big((I+\mathcal{O}_{a_{i-1}}^T\mathcal{O}_{a_i})^{-1}\widehat{C_2\psi_{a_{i-1}}}(-\hat{\psi}_{a_{i-1}}+2I)\mathcal{O}_{a_{i-1}}^T\mathcal{O}_{a_i}\big)^{(A)}\Big)^\vee &- \Delta t\mathcal{O}_{a_i}^{-1}\mathcal{N}_{a_i}\Bigg]\Bigg|_{t=t^j}\\
&+(J_d F_{a_i}^{j-1} - (F_{a_i}^{j-1})^T J_d)^\vee
\end{aligned}
\tag{I.11}
$$

- For the right node of the continuum manipulator $(a_{i=N})$

$$(F^j_{a_N} J_d - J_d (F^j_{a_N})^T)^\vee = -\frac{2\Delta t^2}{l_{\mathcal{I}_N}} \Bigg[\frac{1}{2} C_1 \Big(\mathcal{O}^T_{a_N} \frac{\Delta p_{a_{N-1}}}{l_{\mathcal{I}_N}} - \mathcal{E}_3 \Big) \times \mathcal{O}^T_{a_N} \Delta p_{a_{N-1}} + \frac{1}{l_{\mathcal{I}_N}} \Big(\big((I + \mathcal{O}^T_{a_{N-1}} \mathcal{O}_{a_N})^{-1} C_2 \widehat{\psi_{a_N - 1}} (\hat{\psi}_{a_{N-1}} - 2I) \big)^{(A)} \Big)^\vee - \Delta t \mathcal{O}^{-1}_{a_N} \mathcal{N}_{a_N} \Bigg] \Bigg|_{t=t^j} + (J_d F^{j-1}_{a_N} - (F^{j-1}_{a_N})^T J_d)^\vee \tag{I.12}$$

where the variable $_{a_i}$ is defined as $\hat{\psi}_{a_i} := \exp^{-1}(\mathcal{O}^T_{a_i} \mathcal{O}_{a_{i+1}})$ which is approximated by the Cayley transformation as $\hat{\psi}_{a_i} := \mathrm{Cay}^{-1}(\mathcal{O}^T_{a_i} \mathcal{O}_{a_{i+1}})$, where The Cayley transformation and its inverse are defined in the following form for convenience $\mathcal{O}^T_a \mathcal{O}_{a+1} = \mathrm{Cay}(\hat{\psi}_a) = \frac{I + \hat{\psi}_a}{I - \hat{\psi}_a}$ with inverse $\hat{\psi}_a = \mathrm{Cay}^{-1}(\mathcal{O}^T_a \mathcal{O}_{a+1}) = 2 \frac{\mathcal{O}^T_a \mathcal{O}_{a+1} - I}{\mathcal{O}^T_a \mathcal{O}_{a+1} + I}$ (see, [Dem+15; Lee08]). In addition, $\Delta p_{a_i} \Big|_{t=t^j} = p^{j+1}_{a_i} - p^j_{a_i}$.

For discrete Euler-Lagrange equations for rotations, Equations (I.10)-(I.12), one has to solve an implicit expression of the form

$$\hat{\mathcal{U}} = F_a J_d - J_d F^T_a, \ \forall a \in \{a_0, \cdots, a_N\} \tag{I.13}$$

In order to solve Equation (I.13) for $F \in SO(3)$, (the vector $\mathcal{U}$ or the right hand sides of Equations (I.10)-(I.12) and the symmetric non-standard inertia matrix J_d are known), a Newton iteration method based on the Cayley transformation is applied (as described in [Lee08, Section 3.3.8]) .

I.3.1.2 Discrete Euler-Lagrange Equations for Translations

- For the left node of the continuum manipulator $(a_{i=0})$

$$p^{j+1}_{a_0} = \frac{2\Delta t^2}{l_{\mathcal{I}_0} M} \Bigg[\frac{1}{2} \mathcal{O}_{a_0} C_1 \Big(\mathcal{O}^T_{a_0} \frac{\Delta p_{a_0}}{l_{\mathcal{I}_0}} - \mathcal{E}_3 \Big) + \frac{1}{2} \mathcal{O}_{a_{0+1}} C_1 \Big(\mathcal{O}^T_{a_{0+1}} \frac{\Delta p_{a_0}}{l_{\mathcal{I}_0}} - \mathcal{E}_3 \Big) - \frac{l_{\mathcal{I}_0}}{2} F^c_{a_0} - \Delta t \mathcal{O}^{-1}_{a_0} \mathcal{F}_{a_0} \Bigg] \Bigg|_{t=t^j} + 2p^j_{a_0} + p^{j-1}_{a_0} \tag{I.14}$$

- For the interior nodes of the continuum manipulator $\forall a_i,\ i \in \{1, \cdots, N-1\}$

$$p_{a_i}^{j+1} = \frac{\Delta t^2}{l_{\mathcal{I}_i} M}\Bigg[\frac{1}{2}\mathcal{O}_{a_i} C_1\Big(\mathcal{O}_{a_i}^T \frac{\Delta p_{a_i}}{l_{\mathcal{I}_i}} - \mathcal{E}_3\Big) - \frac{1}{2}\mathcal{O}_{a_{i-1}} C_1\Big(\mathcal{O}_{a_{i-1}}^T \frac{\Delta p_{a_{i-1}}}{l_{\mathcal{I}_i}} - \mathcal{E}_3\Big) + \frac{1}{2}\mathcal{O}_{a_{i+1}} C_1\Big(\mathcal{O}_{a_{i+1}}^T \frac{\Delta p_{a_i}}{l_{\mathcal{I}_i}} - \mathcal{E}_3\Big) - \frac{1}{2}\mathcal{O}_{a_i} C_1\Big(\mathcal{O}_{a_i}^T \frac{\Delta p_{a_{i-1}}}{l_{\mathcal{I}_i}} - \mathcal{E}_3\Big) - \frac{l_{\mathcal{I}_i}}{2} F_{a_i}^c - \Delta t \mathcal{O}_{a_i}^{-1} \mathcal{F}_{a_i}\Bigg]\Bigg|_{t=t^j} + 2p_{a_i}^j + p_{a_i}^{j-1} \tag{I.15}$$

- For the right node of the continuum manipulator $(a_{i=N})$

$$p_{a_N}^{j+1} = \frac{2\Delta t^2}{l_{\mathcal{I}_N} M}\Bigg[- \frac{1}{2}\mathcal{O}_{a_{N-1}} C_1\Big(\mathcal{O}_{a_{N-1}}^T \frac{\Delta p_{a_{N-1}}}{l_{\mathcal{I}_N}} - \mathcal{E}_3\Big) - \frac{1}{2}\mathcal{O}_{a_N} C_1\Big(\mathcal{O}_{a_N}^T \frac{\Delta p_{a_{N-1}}}{l_{\mathcal{I}_N}} - \mathcal{E}_3\Big) - \frac{l_{\mathcal{I}_N}}{2} F_{a_N}^c - \Delta t \mathcal{O}_{a_N}^{-1} \mathcal{F}_{a_N}\Bigg]\Bigg|_{t=t^j} + 2p_{a_N}^j + p_{a_N}^{j-1} \tag{I.16}$$

Remark I.3.1. For magnetic actuation, we fabricate manipulators with embedded permanent magnets. Consider Magnet i with weight m_i in an interval $\mathcal{I}_i$ in which a_i and a_{i+1}, $i \in \{0, \cdots, N-1\}$ are considered as left and right nodes of the interval. Therefore, the distributed load per unit length for Nodes a_i and a_{i+1} are approximately considered as $M + \frac{m_i}{2l_{\mathcal{I}_i}}$. In addition, if Magnet i is embedded at the tip, $M + \frac{m_i}{l_{\mathcal{I}_N}}$ replaces the distributed load per unit length of Node a_{N+1} while the distributed load per unit length of Node a_N is unchanged.

I.3.2 Estimation

In this section, online distributed estimation algorithms are developed to predict the model dissipation error. The structure of the estimator mimics the model's structure, as explained in Section I.3.1. To design the estimation protocol, we follow the same line of ideas as in [Lu95] but in distributed multi-systems configuration. We consider each node as an individual system coupled with the other adjacent nodes, i.e., neighbors, in succession. In other words, each node exchanges its local pose (position and orientation) with its neighbors. It should be noted that the estimation filters are designed and implemented for each node. Figure I.3 shows the configuration of the distributed filters and nodes.

For simplicity, we assume that each node's position is included in its state vector. Therefore, given node a_i, $i = \{0, \cdots, N\}$, the time-varying dynamic equations based on Equations (I.14)-(I.16) can be written as

$$\begin{aligned} \mathtt{S}_{a_i}^{j+1} &= \mathtt{F}_{a_i}(\mathtt{S}_{a_i}^j, \mathtt{U}_{a_i}^j, j) + \mathtt{G}_{a_i}^j(\mathtt{S}_{a_i}^j)\mathcal{F}_{a_i}^j, \\ \mathtt{Y}_{a_i}^{j+1} &= \mathtt{H}_{a_i}^{j+1}\mathtt{S}_{a_i}^{j+1}, \end{aligned} \tag{I.17}$$

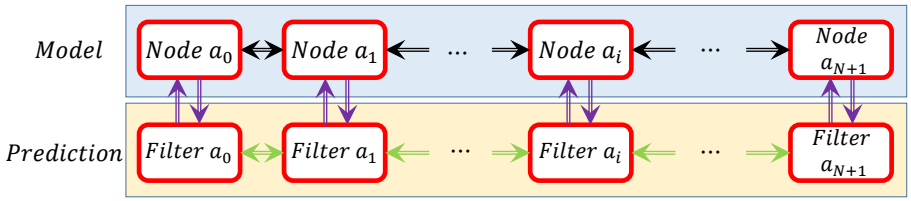


Figure I.3: Configuration of the nodes of the model and the corresponding distributed filters. Filter a_i and Node a_i are coupled with the adjacent nodes in succession.

where $j = \{1, 2, \cdots\}$, $i = \{0, 2, \cdots, N\}$, $\mathtt{S}^j_{a_i} = [p^{j-1}_{a_i}\ p^j_{a_i}]^T \in \mathbb{R}^6$ is the true state vector, $\mathtt{U}^j_{a_i}$ is a known input vector, $\mathtt{F}_{a_i} \in \mathbb{R}^6$ is sufficiently differentiable, $\mathtt{G}^j_{a_i} = [\mathbf{0}_{3\times3}, \frac{-2\Delta t^3}{I_{\mathcal{I}_i} M}\mathcal{O}^j_{a_i}{}^{-1}]^T \in \mathbb{R}^{6\times3}$ is the model dissipation error matrix and in the considered systems is time-varying, $\mathcal{F}^j_{a_i} \in \mathbb{R}^3$ is a modified viscous model dissipation force or hysteretic damping force in the form of $\mathcal{K}^j_{a_i} \circ \Delta t^{-1}(p^j_{a_i} - p^{j-1}_{a_i})$, where $\circ$ denotes Hadamard product and $\mathcal{K}^j_{a_i} \in \mathbb{R}^3$ is damping capacity that is independent of frequency of motion and needs to be estimated, $\mathtt{H}^{j+1}_{a_i} \in \mathbb{R}^{3\times6}$ is the output matrix, and $\mathtt{Y}^{j+1}_{a_i} \in \mathbb{R}^{3\times1}$ is the output vector.

By substituting $\mathcal{F}^j_{a_i} = \mathcal{K}^j_{a_i} \circ \Delta t^{-1}(p^j_{a_i} - p^{j-1}_{a_i})$ into Equation (I.17), one has

$$\begin{aligned} \mathtt{S}^{j+1}_{a_i} &= \mathtt{F}_{a_i}(\mathtt{S}^j_{a_i}, \mathtt{U}^j_{a_i}, j) + \mathtt{G}^j_{a_i}(\mathtt{S}^j_{a_i})\mathcal{K}^j_{a_i} \circ \Delta t^{-1}(p^j_{a_i} - p^{j-1}_{a_i}), \\ \mathtt{Y}^{j+1}_{a_i} &= \mathtt{H}^{j+1}_{a_i}\mathtt{S}^{j+1}_{a_i}, \end{aligned} \tag{I.18}$$

Using commutative property of Hadamard product, we can write $\mathtt{G}^j_{a_i}(\mathtt{S}^j_{a_i})\mathcal{K}^j_{a_i} \circ \Delta t^{-1}(p^j_{a_i} - p^{j-1}_{a_i}) = \mathtt{G}^j_{a_i}(\mathtt{S}^j_{a_i})\Delta t^{-1}(p^j_{a_i} - p^{j-1}_{a_i}) \circ \mathcal{K}^j_{a_i}$. Then, Hadamard product can be converted to matrix multiplication by the corresponding diagonal matrix of the vector $\mathtt{G}^j_{a_i}(\mathtt{S}^j_{a_i})\Delta t^{-1}(p^j_{a_i} - p^{j-1}_{a_i})$ which is denoted by $\mathcal{G}^j_{a_i} = \mathtt{G}^j_{a_i}(\mathtt{S}^j_{a_i})\Delta t^{-1}\mathrm{diag}\big(p^j_{a_i} - p^{j-1}_{a_i}\big)$ and $\mathcal{G}^j_{a_i} \in \mathbb{R}^{6\times3}$. Therefore, Equation (I.18) may be written as

$$\begin{aligned} \mathtt{S}^{j+1}_{a_i} &= \mathtt{F}_{a_i}(\mathtt{S}^j_{a_i}, \mathtt{U}^j_{a_i}, j) + \mathcal{G}^j_{a_i}(\mathtt{S}^j_{a_i})\mathcal{K}^j_{a_i}, \\ \mathtt{Y}^{j+1}_{a_i} &= \mathtt{H}^{j+1}_{a_i}\mathtt{S}^{j+1}_{a_i}, \end{aligned}$$

Estimation of the state and output vector is given by

$$\begin{aligned} \hat{\mathtt{S}}^{j+1}_{a_i} &= \mathtt{F}_{a_i}(\hat{\mathtt{S}}^j_{a_i}, \mathtt{U}^j_{a_i}, j) + \mathcal{G}^j_{a_i}(\hat{\mathtt{S}}^j_{a_i})\hat{\mathcal{K}}^j_{a_i}, \\ \hat{\mathtt{Y}}^{j+1}_{a_i} &= \mathtt{H}^{j+1}_{a_i}\hat{\mathtt{S}}^{j+1}_{a_i}, \end{aligned}$$

where $\hat{\mathtt{S}}^j_{a_i} = [\hat{p}^{j-1}_{a_i}\ \hat{p}^j_{a_i}]^T \in \mathbb{R}^6$ is the estimation of the state vector, $\hat{\mathcal{K}}^j_{a_i} \in \mathbb{R}^3$ is the model dissipation error estimates, $\hat{\mathtt{Y}}^{j+1}_{a_i} \in \mathbb{R}^{3\times1}$ is the output vector estimates. Finally, $\tilde{\mathtt{Y}}^{j+1}_{a_i}$ denotes the measurement. The block diagram of the filter integrated with the model is shown in Figure I.4. To find $\hat{\mathcal{K}}^j_{a_i}$ for the node a_i at the time j, we consider a pointwise cost function that penalizes and

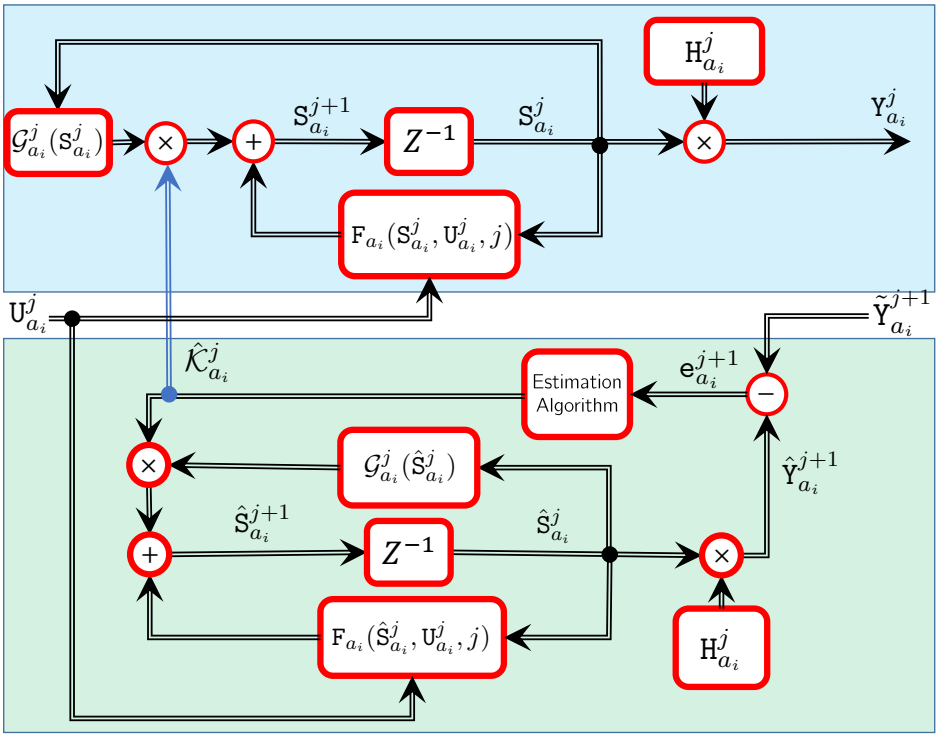


Figure I.4: Block diagram of the proposed prediction filter a_i coupled with the node a_i's model. The filter employs the model's output to perform a pointwise optimization problem to predict the damping capacity $\hat{\mathcal{K}}^i_{a_i}$.

minimizes the estimation error vector (the error between measurement and the output estimation) at the next sampling time $j+1$, and estimated damping capacity $\mathcal{K}^{j+1}_{a_i}$. The cost function for each node a_i is given as

$$J_{a_i}(\mathcal{K}^{j+1}_{a_i}) = \frac{1}{2}\mathtt{e}^{j+2^T}_{a_i}\mathscr{R}\mathtt{e}^{j+2}_{a_i} + \frac{1}{2}\mathcal{K}^{j+1^T}_{a_i}\mathscr{W}\mathcal{K}^{j+1}_{a_i} \tag{I.19}$$

where $\mathtt{e}^{j+2}_{a_i} = \hat{\mathtt{Y}}^{j+2}_{a_i} - \tilde{\mathtt{Y}}^{j+2}_{a_i}$, $\mathscr{W} \in \mathbb{R}^{3\times3}$, and $\mathscr{R} \in \mathbb{R}^{3\times3}$ are positive semi-definite and positive definite matrices, respectively.

In order to derive an optimal estimation law, we need to approximate the output estimation vector $\hat{\mathtt{Y}}^{j+2}_{a_i}$ at the next sampling time $j+2$, which is given by its Taylor series expansion as follows

$$\hat{\mathtt{Y}}^{j+2}_{a_i} \approx \hat{\mathtt{Y}}^{j+1}_{a_i} + \mathtt{Z}(\hat{\mathtt{S}}^{j+1}_{a_i}, \Delta t) + \Lambda(\Delta t)\mathtt{M}(\hat{\mathtt{S}}^{j+1}_{a_i})\hat{\mathcal{K}}^{j+1}_{a_i} \tag{I.20}$$

where

$$\begin{aligned}\mathtt{Z}(\hat{\mathtt{S}}_{a_i}^{j+1}, j) &= \Delta t L^1_{\mathtt{F}_{a_i}}(\mathtt{H}_{a_i}^{j+1}\hat{\mathtt{S}}_{a_i}^{j+1}) = \Delta t \frac{\partial \mathtt{H}_{a_i}^{j+1}\hat{\mathtt{S}}_{a_i}^{j+1}}{\partial \hat{\mathtt{S}}_{a_i}^{j+1}}\mathtt{F}_{a_i}\\ \mathtt{\Lambda}(\Delta t) &= \Delta t I_3\\ \mathtt{M}(\hat{\mathtt{S}}_{a_i}^{j+1}) &= \frac{-2\Delta t^2}{l_{\mathcal{I}_i} M}{\mathcal{O}_{a_i}^j}^{-1}\mathrm{diag}\big(p_{a_i}^j - p_{a_i}^{j-1}\big)\end{aligned}$$

Similarly, we may expand the i^{th} component of $\tilde{\mathtt{Y}}_{a_i}^{j+2}$ in an first-order Taylor series so that

$$\tilde{\mathtt{Y}}_{a_i}^{j+2} \approx \tilde{\mathtt{Y}}_{a_i}^{j+1} + \mathtt{d}_{a_i}^{j+1}$$

where the h^{th} component of $\mathtt{d}_{a_i}^{j+1} \in \mathbb{R}^3$ is

$$\mathtt{d}_{a_i}^{j+1}{}_h = \tilde{\mathtt{Y}}_{a_i}^{j+1}{}_h - \tilde{\mathtt{Y}}_{a_i h}^{j}$$

Solving Equation (I.19) for $\hat{\mathcal{K}}_{a_i}^{j+1}$ by considering Equation (I.20) yields

$$\begin{aligned}\hat{\mathcal{K}}_{a_i}^{j+1} = \Big\{[\mathtt{\Lambda}(\Delta t)\mathtt{M}(\hat{\mathtt{S}}_{a_i}^{j+1})]^T\mathscr{R}[\mathtt{\Lambda}(\Delta t)\mathtt{M}(\hat{\mathtt{S}}_{a_i}^{j+1})] + \mathcal{W}\Big\}^{-1} &\times [\mathtt{\Lambda}(\Delta t)\mathtt{M}(\hat{\mathtt{S}}_{a_i}^{j+1})]^T\mathscr{R}\\ &\times [\mathtt{Z}(\hat{\mathtt{S}}_{a_i}^{j+1}, j) + \mathtt{e}_{a_i}^{j+1} - \mathtt{d}_{a_i}^{j+1}]\end{aligned} \tag{I.21}$$

Stability and convergence analysis of the filters can be found in [Lu95]. Here we skip the analysis for brevity.

I.4 Simulation and Experimental Results

In this section, we investigate and analyze the solver's performance with different continuum manipulators through experiments. The experiments here are expected to provide validation of the theoretical formulation for a variety of scenarios. As discussed earlier, it is worth remembering that the dynamic equations for translation and rotation are decoupled. Equations (I.14)-(I.16) can be solved explicitly to update nodes translation, while an iterative method — as it is discussed in Section I.3.1.1 — is necessary to solve Equations (I.10)-(I.12) for updating the rotations. It should be pointed out that the estimation law (I.21) is implemented for every node to estimate conservative forces. The required parameters for the simulation will be discussed for each experiment.

I.4.1 Flexible Metal Rods

As a first case, we consider a cylindrical rod made of aluminum (Al4043/ AlSi5) with diameter of 2 mm, length 200 mm, mass density 2690 $\frac{\mathrm{kg}}{\mathrm{m}^3}$, Young's modulus 75 GPa and Poisson's ratio 0.33. As a first example, we suppose a planar motion of the rod in the $\mathcal{E}_1\mathcal{E}_3$-plane with the initial deflection $\theta_{\mathcal{E}_1\mathcal{E}_3} = 3.69^\circ$. $\theta_{\mathcal{E}_1\mathcal{E}_3}$ denotes

Table I.3: Simulation parameters in Equations (I.10)-(I.16) and (I.21) for in-plane experiment of flexible cylindrical rod (AlSi05).

M	$8.45 \times 10^{-3}\ \frac{\text{g}}{\text{mm}}$
Number of elements	15
$l_{\mathcal{I}_i}\big\vert_{i=\{1,2,\cdots,N\}}$	$\frac{200}{15}$ mm
J_d	$\text{diag}(0, 2.11, 2.11)$ g $\times$ mm^2
$\mathcal{E}_3$	$[1, 0, 0]^T$
$F^c_{a_i}\big\vert_{i=\{1,2,\cdots,N+1\}}$	$[0, 0, 8.29]^T \times 10^4\ \frac{\text{g}}{\text{S}^2}$
C_1	$\text{diag}(2.35, 0.88, 0.88) \times 10^{14}$
C_2	$\text{diag}(4.42, 5.89, 5.89) \times 10^{13}$
Time step	1×10^{-6}
Simulation time	1.5 (S)
$\mathscr{R}$	$I_3 \times 10^4$
$\mathscr{W}$	$I_3 \times 10^{-1}$

the rotation of the tip around $\mathcal{E}_1$ in the $\mathcal{E}_1\mathcal{E}_3$-plane. It is worth pointing out that the nondissipative force is the gravity in the $\mathcal{E}_3$-axis direction. In addition, we run a simulation with the given specifications with $N = 15$ discretization nodes. These points are depicted in Figure I.5 together with some time-evolved configurations of the rod. For simplicity, only tip positions are used for the comparison with the simulation results. The maximum and mean absolute error are 0.15 mm (i.e., 2.5% of displacement), and 0.05 mm, respectively. The error, simulation and experiment results are shown in Figure I.6 and the simulation parameters are summarized in Table I.3.

Next, we consider a three-dimensional motion for a rod with the same material as the first case but with diameter $d = 1$ mm with initial deflections $\theta_{\mathcal{E}_1\mathcal{E}_2} = -5.53^\circ$ (i.e., the tip distance is 8 mm from $\mathcal{E}_2$-axis) and $\theta_{\mathcal{E}_1\mathcal{E}_3} = 6.52^\circ$ (i.e., the tip distance is 10 mm from $\mathcal{E}_3$-axis) in the $\mathcal{E}_3\mathcal{E}_2$ and $\mathcal{E}_1\mathcal{E}_2$ planes, respectively. The results of the experiment, simulation, and error are depicted in Figure I.7. Maximum and mean absolute error in both $\mathcal{E}_3$ and $\mathcal{E}_1$ axes are 0.15 mm (i.e., 2.12%), and 0.05 mm, respectively. The simulation parameters are summarized in Table I.4.

I.4.2 Polymer-based Rods

In the second experiment, a cylindrical Polydimethylsiloxane (PDMS) rod is considered. Figure I.8 depicts the rod, which has diameter $D = 5$ mm, length $L = 60.5$ mm. In addition, for the rod, mass density $\rho = 1101\ \frac{\text{kg}}{\text{m}^3}$, Young's modulus $E = 365.12$ MPa, and Poisson's ratio $\nu = 0.5$. In this experiment, the rod is kept straight initially, with the gravity acting along $\mathcal{E}_2$, Figure I.8(a). Experiment and simulation results for the tip position and the error are depicted in Figure I.9. Also, maximum and mean absolute error in $\mathcal{E}_2$-axis are 0.56 mm

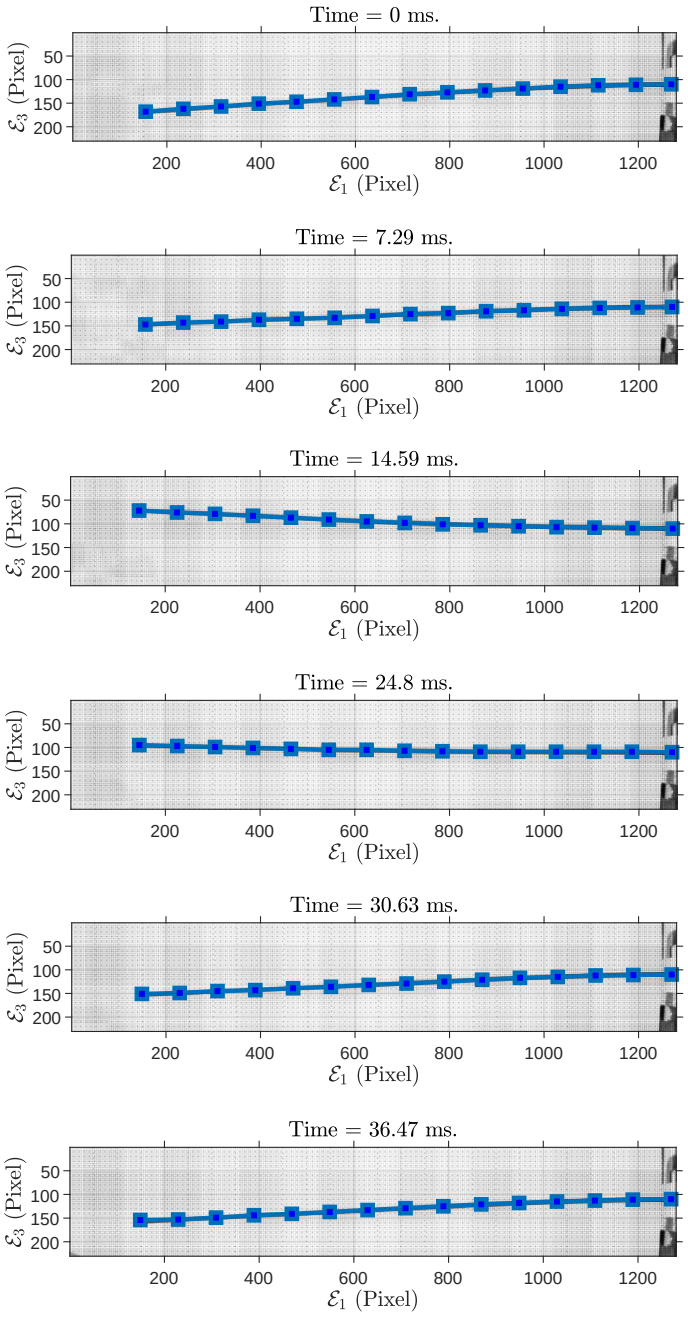
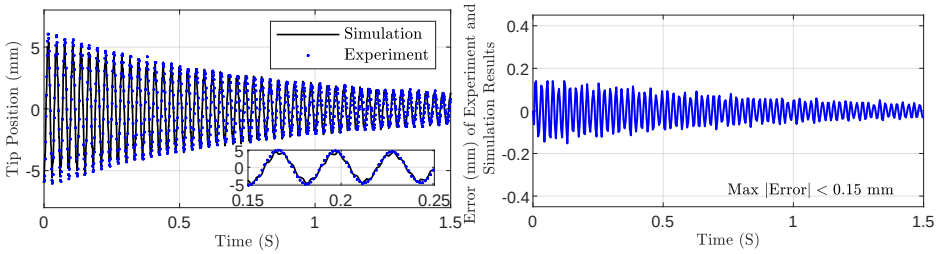


Figure I.5: Sample of grabbed images of flexible rod (AlSi05) configurations, in-plane experiment ($\mathcal{E}_1\mathcal{E}_3$-plane). 15 discretization nodes, depicted in blue squares, are superimposed on the flexible rod.



(a) (b)

Figure I.6: Simulation and experiment results for flexible rod (AlSi05), in-plane experiment: (a) Tip position in $\mathcal{E}_1$-axis direction. Inset highlights the results in a small time range. (b) Error in $\mathcal{E}_1$-axis direction.

Table I.4: Simulation parameters in Equations (I.10)-(I.16) and (I.21) for out-of-plane experiment of flexible cylindrical rod (AlSi05).

M	2.11×10^{-3} $\frac{\text{g}}{\text{mm}}$
Number of elements	15
$l_{\mathcal{I}_i}\big\vert_{i=\{1,2,\cdots,N\}}$	$\frac{200}{15}$ mm
J_d	$\text{diag}(0, 0.13, 0.13)$ g $\times$ mm^2
$\mathcal{E}_3$	$[1, 0, 0]^T$
$F^c_{a_i}\big\vert_{i=\{1,2,\cdots,N+1\}}$	$[0, 0, 2.07]^T \times 10^4$ $\frac{\text{g}}{\text{S}^2}$
C_1	$\text{diag}(5.89, 2.21, 2.21) \times 10^{13}$
C_2	$\text{diag}(2.76, 3.68, 3.68) \times 10^{12}$
Time step	1×10^{-6}
Simulation time	1.5 (S)
$\mathscr{R}$	$I_3 \times 10^4$
$\mathscr{W}$	$I_3 \times 10^{-1}$

(i.e., 4.87%), and 0.05 mm, respectively. In $\mathcal{E}_3$-axis, maximum and mean absolute error are 0.28 mm (i.e., 4.89%), and 0.05 mm, respectively. The simulation parameters are summarized in Table I.5.

For the next experiment, we fabricated a cylindrical PDMS manipulator with a permanent magnet at the tip. The initial and some time-evolved configurations of the rod are depicted in Figure I.10. The specifications of the rod are as follows: diameter $D = 4$ mm, length $L = 60.55$ mm. In addition, the embedded neodymium magnet is a cylindrical magnet with diameter $D_m = 2$ mm, height $L_m = 4$ mm, mass $M_m = 9.6 \times 10^{-5}$ kg. The rod moves around $\mathcal{E}_1$ with the initial deflection $\theta_{\mathcal{E}_2\mathcal{E}_3} = -41.74$ in the $\mathcal{E}_2\mathcal{E}_3$-plane. Also, maximum and mean absolute error in $\mathcal{E}_3$-axis are 0.55 mm (i.e., 1.16%), and 0.13 mm, respectively. In $\mathcal{E}_2$-axis, maximum and mean absolute error are 0.61 mm (i.e., 3.35%), and 0.14 mm, respectively. Tip positions in the experiment, simulation and the error are

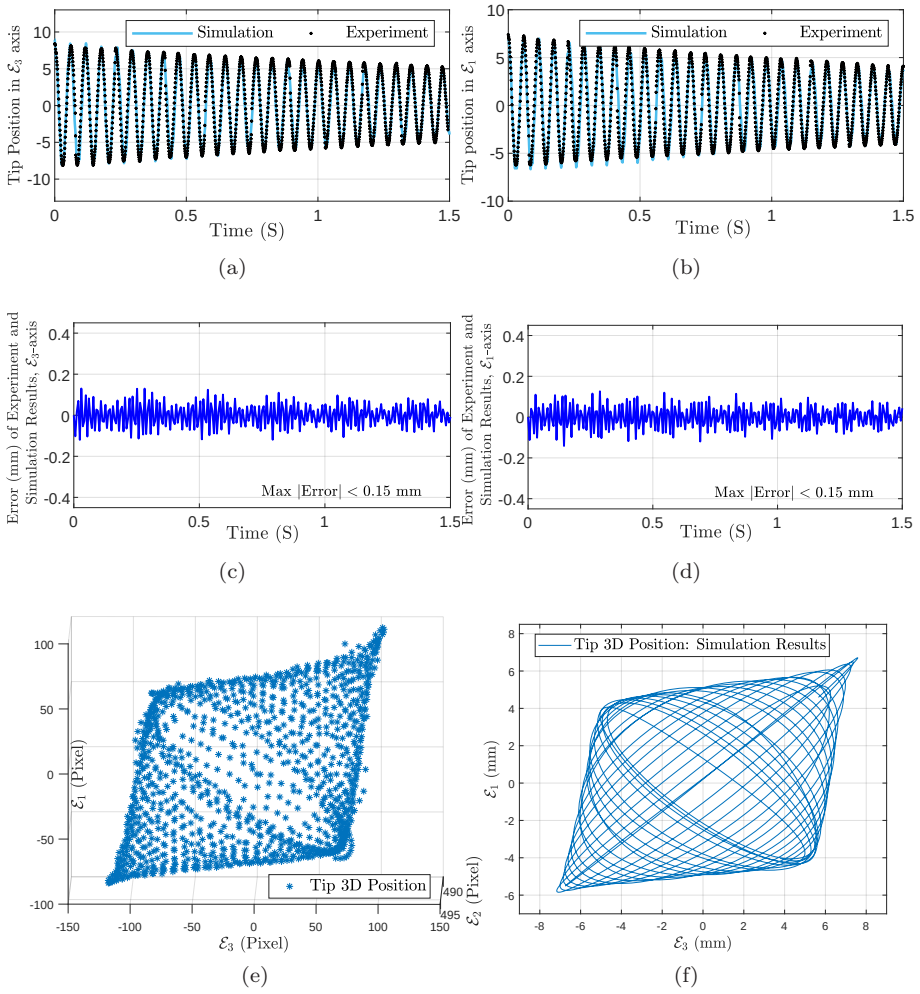


Figure I.7: Simulation and experiment results for flexible rod (AlSi05), out-of-plane experiment: (a) Tip position in $\mathcal{E}_3$-axis direction. (b) Tip position in $\mathcal{E}_1$-axis direction. (c) Error in $\mathcal{E}_3$-axis direction. (d) Error in $\mathcal{E}_1$-axis direction. (e) Tip 3D position: non-planar experiment. (f) Tip 3D position: simulation.

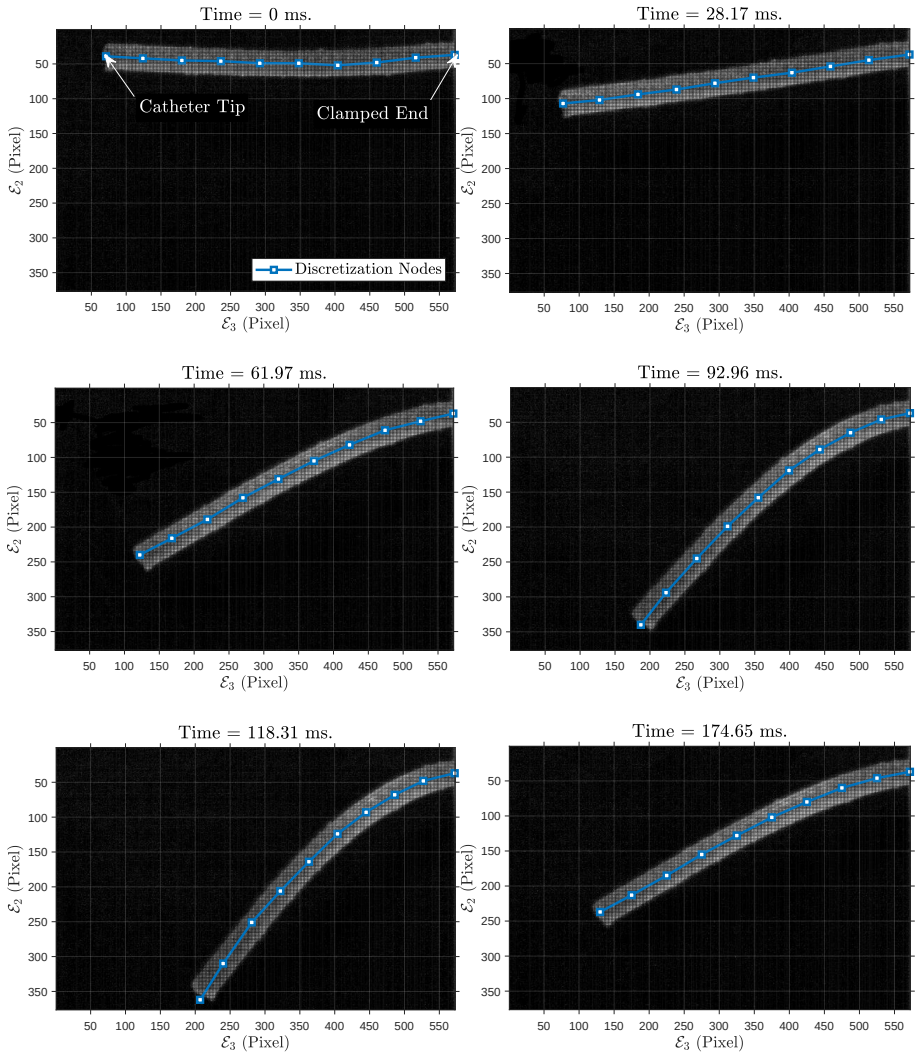


Figure I.8: Sample of grabbed images for Polydimethylsiloxane (PDMS) rod without any embedded magnet in 2D experiment: in-plane motion. Also, 10 discretization points are superimposed on the soft rod.

Table I.5: Simulation parameters in Equations (I.10)-(I.16) and (I.21) for in-plane experiment of PDMS rod (without magnet).

M	21.62×10^{-3} $\frac{\text{g}}{\text{mm}}$
Number of elements	10
$l_{\mathcal{I}_i}\big\vert_{i=\{1,2,\cdots,N\}}$	$\frac{60.5}{10}$ mm
J_d	$\text{diag}(0, 33.78, 33.78)$ g $\times$ mm^2
$\mathcal{E}_3$	$[1, 0, 0]^T$
$F^c_{a_i}\big\vert_{i=\{1,2,\cdots,N+1\}}$	$[0, 0, 2.12]^T \times 10^5$ $\frac{\text{g}}{\text{S}^2}$
C_1	$\text{diag}(7.17, 2.39, 2.39) \times 10^9$
C_2	$\text{diag}(0.75, 1.12, 1.12) \times 10^{10}$
Time step	8×10^{-5}
Simulation time	1.5 (S)
$\mathscr{R}$	$I_3 \times 10^5$
$\mathscr{W}$	$I_3 \times 10^{-1}$

shown in Figure I.11. The simulation parameters are summarized in Table I.6.

Table I.6: Simulation parameters in Equations (I.10)-(I.16) and (I.21) for in-plane and circular-motion experiments of PDMS rod (with tip magnet).

M	13.83×10^{-3} $\frac{\text{g}}{\text{mm}}$
Number of elements	10
$l_{\mathcal{I}_i}\big\vert_{i=\{1,2,\cdots,N\}}$	$\frac{60.55}{10}$ mm
J_d	$\text{diag}(0, 13.83, 13.83)$ g $\times$ mm^2
$\mathcal{E}_3$	$[1, 0, 0]^T$
$F^c_{a_i}\big\vert_{i=\{1,2,\cdots,N\}}$	$[0, 0, 1.357]^T \times 10^5$ $\frac{\text{g}}{\text{S}^2}$
$F^c_{a_{N+1}}$	$[0, 0, 1.36]^T \times 10^5$ $\frac{\text{g}}{\text{S}^2}$
C_1	$\text{diag}(4.58, 1.53, 1.53) \times 10^9$
C_2	$\text{diag}(3.06, 4.59, 4.59) \times 10^9$
Time step	9×10^{-5}
Simulation time	2 (S)
$\mathscr{R}$	$I_3 \times 10^5$
$\mathscr{W}$	$I_3 \times 10^{-1}$
Magnet weight	0.096 g

Hereafter, a magnetic field generation setup is employed to actuate the manipulators. The following section introduces magnetic field generation setup and the related background.

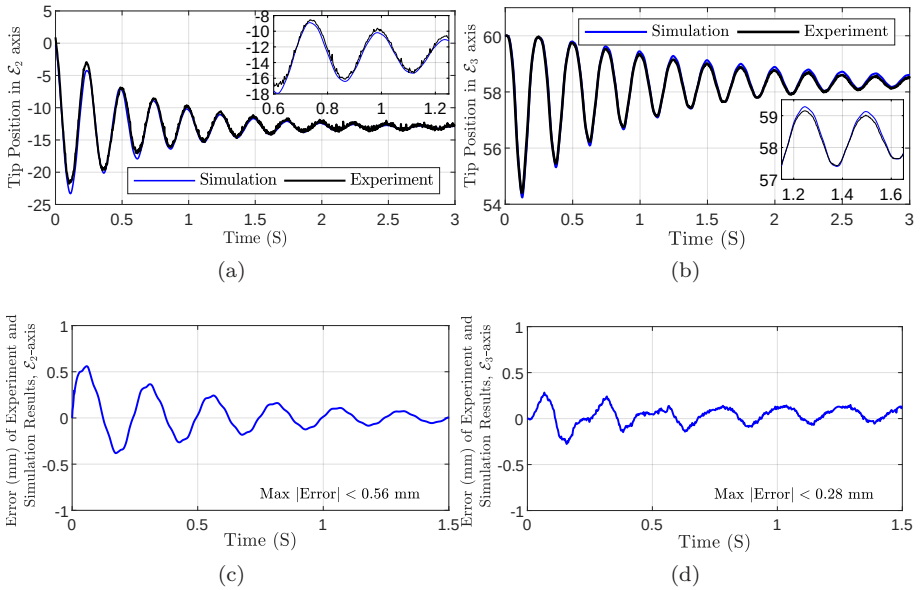


Figure I.9: Simulation and experiment results of Polydimethylsiloxane (PDMS) rod, planar motion: (a) Tip Position in $\mathcal{E}_2$-axis direction. Inset magnifies the results in a small time range. (b) Tip Position in $\mathcal{E}_3$-axis direction. (c) Error in $\mathcal{E}_2$-axis direction. (d) Error in $\mathcal{E}_3$-axis direction.

I.4.3 Magnetic Field Generation

The setup used here consists of two pairs of Helmholtz coils to generate magnetic fields. Each pair consists of two identical electromagnetic coils, as shown in Figure I.12. The first pair of coils generates a uniform magnetic field along the $\mathcal{E}_1$-axis. The second pair of smaller coils are placed inside the first pair to produce a field along the $\mathcal{E}_2$-axis. Two cameras are placed next to the setup to monitor the side view of the workspace. For image acquisition, we use both cameras in a stereo vision setup to reconstruct 3D views of the manipulator's motion.

The setup produces a maximum magnetic field $B_u = 45$ mT.

For the first experiment using magnetic actuation, we use the rod with a neodymium magnet with diameter 2 mm, height 4 mm, and magnetisation N45.

First, a magnetic field $B_g = 7.75$ mT is applied to compensate for the gravity. Then, the tip of the manipulator is induced to rotate in a circle in the $\mathcal{E}_2\mathcal{E}_3$-plane using a rotating magnetic field of magnitude $B_u = 14.5$ mT.

The magnetic field produces force and torque F_{rot} and τ_{rot}, respectively, given by

$$F_{rot} = \nabla(m \cdot B_{rot}),$$
$$\tau_{rot} = m \times B_{rot}$$

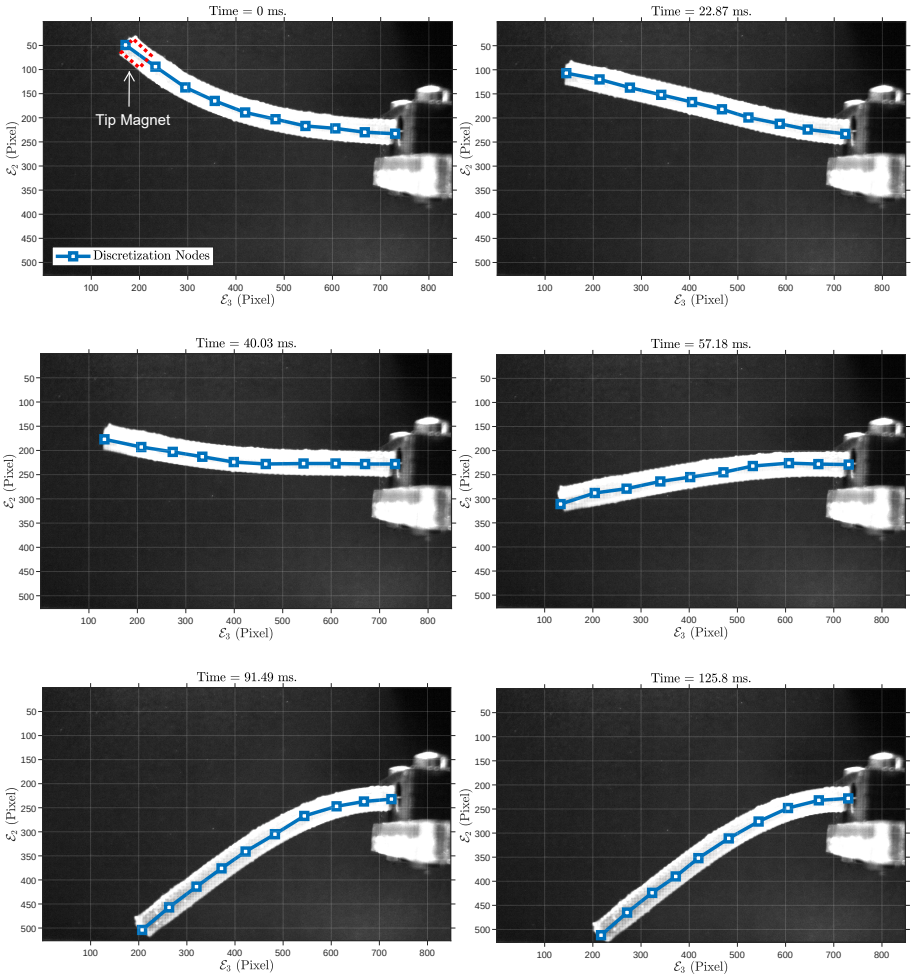


Figure I.10: Sample of grabbed images: Polydimethylsiloxane (PDMS) rod with an embedded magnet at the tip in 2D experiment: motion in a plane. Also, 10 discretization points are shown on the soft rod.

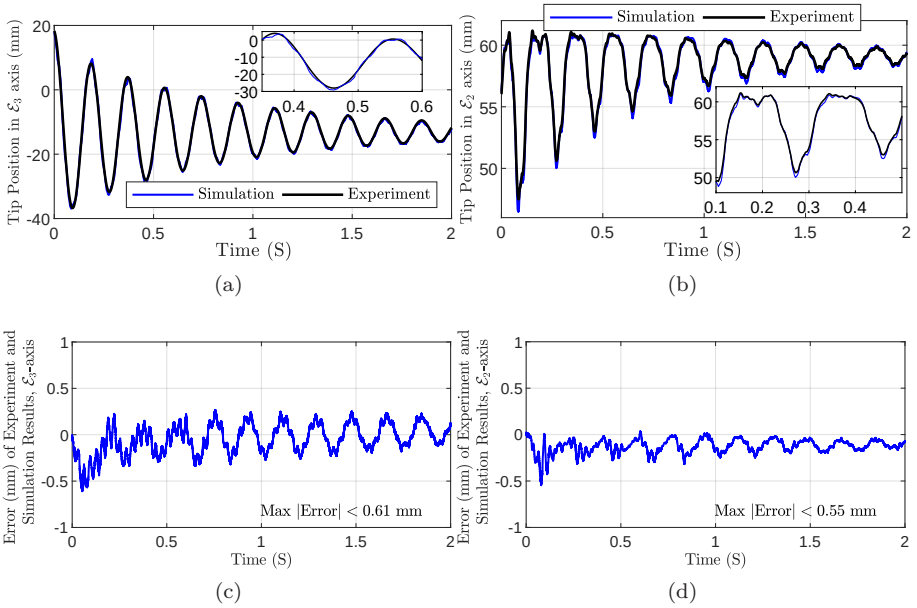


Figure I.11: Simulation and experiment results of Polydimethylsiloxane (PDMS) rod with an embedded magnet at the tip, planar experiment: (a) Tip Position in $\mathcal{E}_3$-axis direction. Inset highlights the results in a small time range. (b) Tip Position in $\mathcal{E}_2$-axis direction. (c) Error in $\mathcal{E}_3$-axis direction. (d) Error in $\mathcal{E}_2$-axis direction.

where m is the dipole moment of the tip's magnet. The dipole moment can be computed as $m = \frac{1}{\mu_0} B_r V$ in which residual magnetism $B_r \in [1.32, 1.37]$ mT, μ_0 is the permeability of vacuum, and the volume of the magnet, $V = 4\pi$ mm^3. Experiment, simulation results and the error are shown in Figure I.13. It should be noted that the error plot shows Euclidean norm of the tip position in the experiment and simulation. Also, the maximum and mean absolute errors are 1.20 mm (i.e., 1.43%) and 0.59 mm, respectively. Since we use the same manipulator as the previous experiment, simulation parameters can be found in Table I.6.

As a last experiment, we fabricated a PDMS continuum manipulator with a square cross-section and two embedded permanent magnets, one at the tip and another in the middle — 36.1 mm from the tip — of the manipulator. The embedded neodymium magnets are identical cylindrical magnets with different dipole moment's directions and diameter $D_m = 2$ mm, height $L_m = 3$ mm, weight $M_m = 7.2 \times 10^{-5}$ kg. The embedded magnets are induced to pursue a prescribed motion in the $\mathcal{E}_2\mathcal{E}_3$-plane using a varying magnetic field of initial and final magnitude $B_u = 20$ mT and 19.85 mT. The initial and some time-evolved configurations of the rod are depicted in Figure I.14. It should be

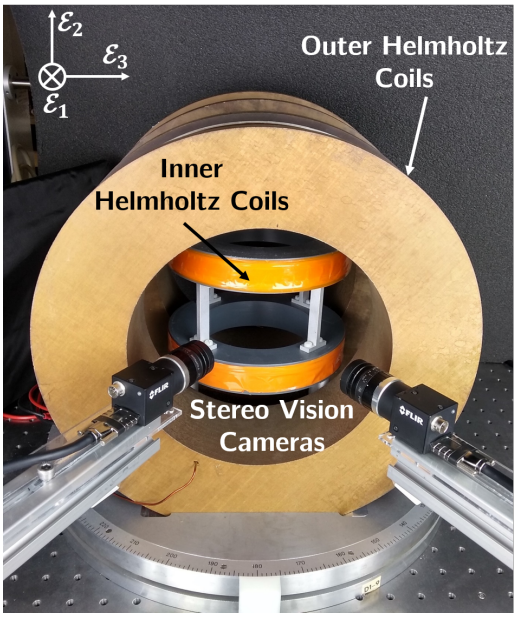


Figure I.12: Magnetic field generation setup with the stereo vision cameras. Two nested pairs of Helmholtz coils generate uniform magnetic fields in $\mathcal{E}_1$ and $\mathcal{E}_2$-axes direction.

noted that analysis of magnetic force and torque follows the same procedure as described above. The specifications of the manipulator are as follows: edge length $a = 2$ mm, length $L = 85.5$ mm. The maximum and mean absolute error for the tip magnet are 1.00 mm (i.e., 2.24%) and 0.15 mm, in $\mathcal{E}_2$-axis direction, respectively. In $\mathcal{E}_3$-axis direction, the maximum and mean absolute error for the tip magnet are 1.40 mm (i.e., 5.13%) and 0.33 mm, respectively. The maximum and mean absolute error for the middle magnet are 0.47 mm (i.e., 2.05%) and 0.08 mm, in $\mathcal{E}_2$-axis direction, respectively. In $\mathcal{E}_3$-axis direction, the maximum and mean absolute error for the middle magnet are 0.40 mm (i.e., 3.68%) and 0.10 mm, respectively.

The position of the tip and middle magnets in the experiment and simulation and also the error is shown in Figure I.15. For this experiment, the simulation parameters are summarized in Table I.7.

I.5 Discussion

We validate our approach by designing and carrying out different experiments with flexible metal rods and polymer-based soft rods. The results are summarized in Table I.8.

Table I.8 demonstrates the maximum and the mean absolute values of the errors. As we observe from this table, the simulation results closely match

Table I.7: Simulation parameters in (Equation (I.10))-(Equation (I.16)) and (Equation (I.21)) for in-plane experiment of square cross-section PDMS rod (with 2 magnets).

M	4.40×10^{-3} $\frac{\text{g}}{\text{mm}}$
Number of elements	10
$l_{\mathcal{I}_i}\big\vert_{i=\{1,2,\cdots,N\}}$	$\frac{85.5}{10}$ mm
J_d	$\text{diag}(0, 1.47, 1.47)$ g $\times$ mm^2
$\mathcal{E}_3$	$[1, 0, 0]^T$
$F^c_{a_i}\big\vert_{i=\{1,2,5,\cdots,N\}}$	$[0, 0, 4.32]^T \times 10^4$ $\frac{\text{g}}{\text{S}^2}$
$F^c_{a_{N+1}} \approx F^c_{a_{3,4}}$	$[0, 0, 4.33]^T \times 10^4$ $\frac{\text{g}}{\text{S}^2}$
C_1	$\text{diag}(1.46, 0.49, 0.49) \times 10^{10}$
C_2	$\text{diag}(3.25, 4.87, 4.87) \times 10^9$
Time step	1×10^{-4}
Simulation time	7.5 (S)
$\mathscr{R}$	$I_3 \times 10^5$
$\mathscr{W}$	$I_3 \times 10^{-1}$
Tip and middle magnets weight	0.072 g

the experimental responses, i.e., for Experiments 1 and 2 in which the flexible metal rods (AlSi05) are employed, the worst-case errors are $< 0.01\%$ of the manipulators' length. For dynamic Experiments 3 and 4 in which the PDMS rods are used, maximum of errors respectively are 0.95% and 1% of the manipulator's length. For the polymer rods, higher errors are due to the uncertainties in fabrication and nonlinear elastic properties. In quasi-static Experiments 5 and 6, the manipulators experience large deformations and external loads; the worst-case errors are less than 2% and 1% of the manipulators' length. It should be pointed out that compared to the manipulators' length, the mean absolute deviations are small, which reflect the model's performance.

During the implementation of the modeling approach, it was observed that the number of nodes affects the frequency of motion. Increasing the number of nodes provides a more accurate solution for the frequency of the system. However, the computation time increases significantly with the number of nodes. Therefore, to be able to run the simulations in a reasonable amount of time and with a small number of nodes, frequency shaping was necessary to be able to match the results.

The following example shows the motivation behind the frequency shaping of the motion. Consider a manipulator with the following specifications: Length $L = 0.5$ m, mass density $\rho = 1000$ $\frac{\text{kg}}{\text{m}^3}$, square cross-section with edge length $a = 5$ cm, Poisson's ratio $\nu = 0.35$, and Young's modulus $E_{hf} = 5 \times 10^4$ KPa in the high frequency case and $E_{lf} = 500$ KPa. Figure I.16 compares the position between high and low-frequency cases. The base of the manipulator is fixed at the origin. From Figure I.16, it is observed that by changing the Young Modulus from E_{hf} to E_{lf}, tip motion is preserved but in a scaled frequency.

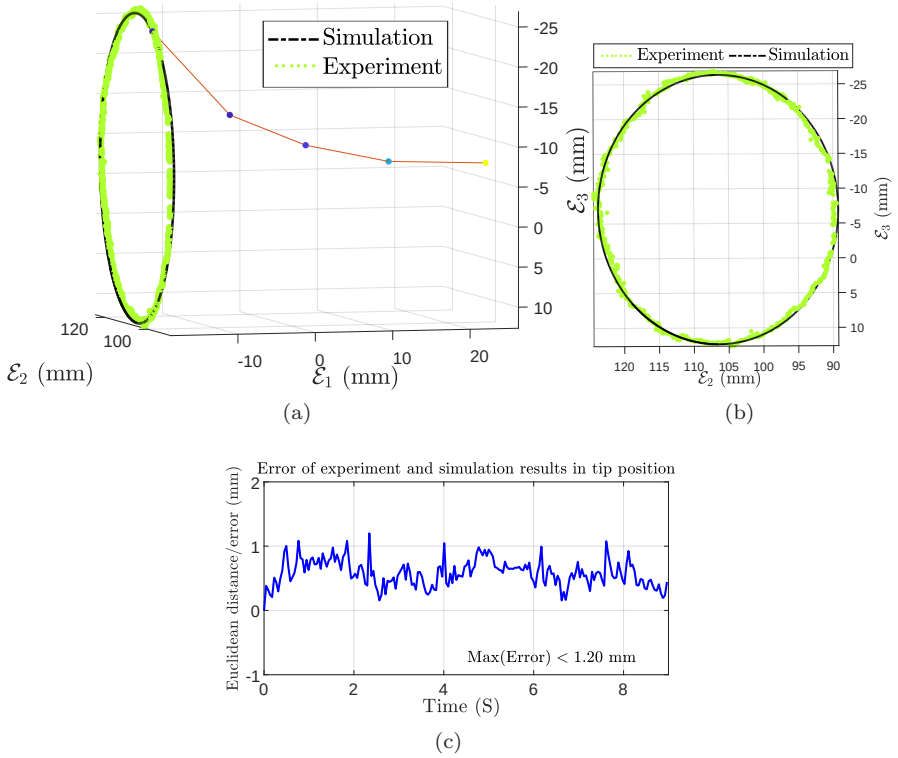


Figure I.13: Reconstruction of the scene for circular motion of Polydimethylsiloxane (PDMS) rod with an embedded magnet at the tip: (a,b) 3D experiment and simulation results. (c) Euclidean distance/error of experiment and simulation results in tip position.

The frequency of the continuum motion is only dependant on parameters such as length, the moment of inertia of the cross-section, Young modulus, and material density. Then, the natural frequency of the continuum manipulator with a fixed end and free tip can be written as

$$\omega_{nf} \propto \sqrt{\frac{EI}{\rho A L^4}}$$

Consider a manipulator with an equivalent spatial discretization of its central line by $\bar{N}$ elements. $\omega_{nf_{\bar{N}}}$ denotes the natural frequency of the manipulator in the simulation with $\bar{N}$ elements, and one has

$$\omega_{nf_{\bar{N}}} \propto \sqrt{\frac{\gamma_{corr} EI}{\rho A L^4}}$$

The correction factor which needs to be multiplied by Young Modulus in the

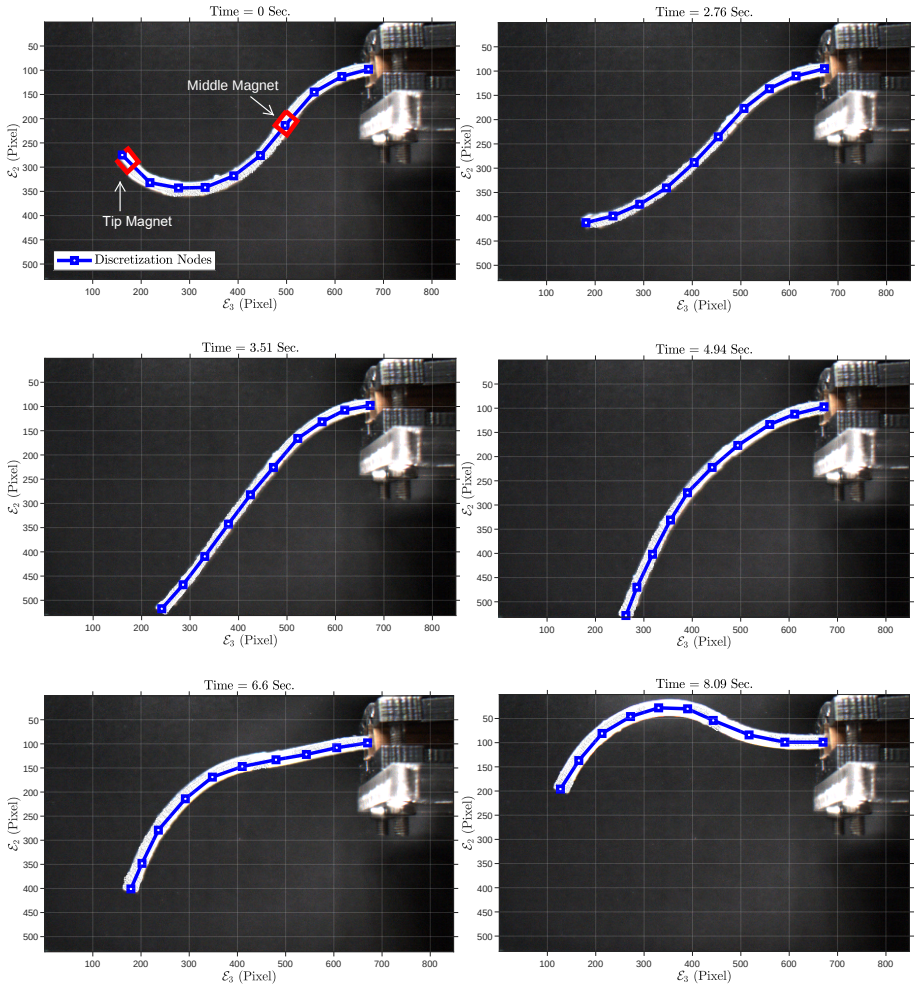


Figure I.14: Sample of grabbed images: Polydimethylsiloxane (PDMS) rod with two embedded magnets, 2D experiment: motion in a plane. Also, 10 discretization points, (blue squares,) are shown on the rod.

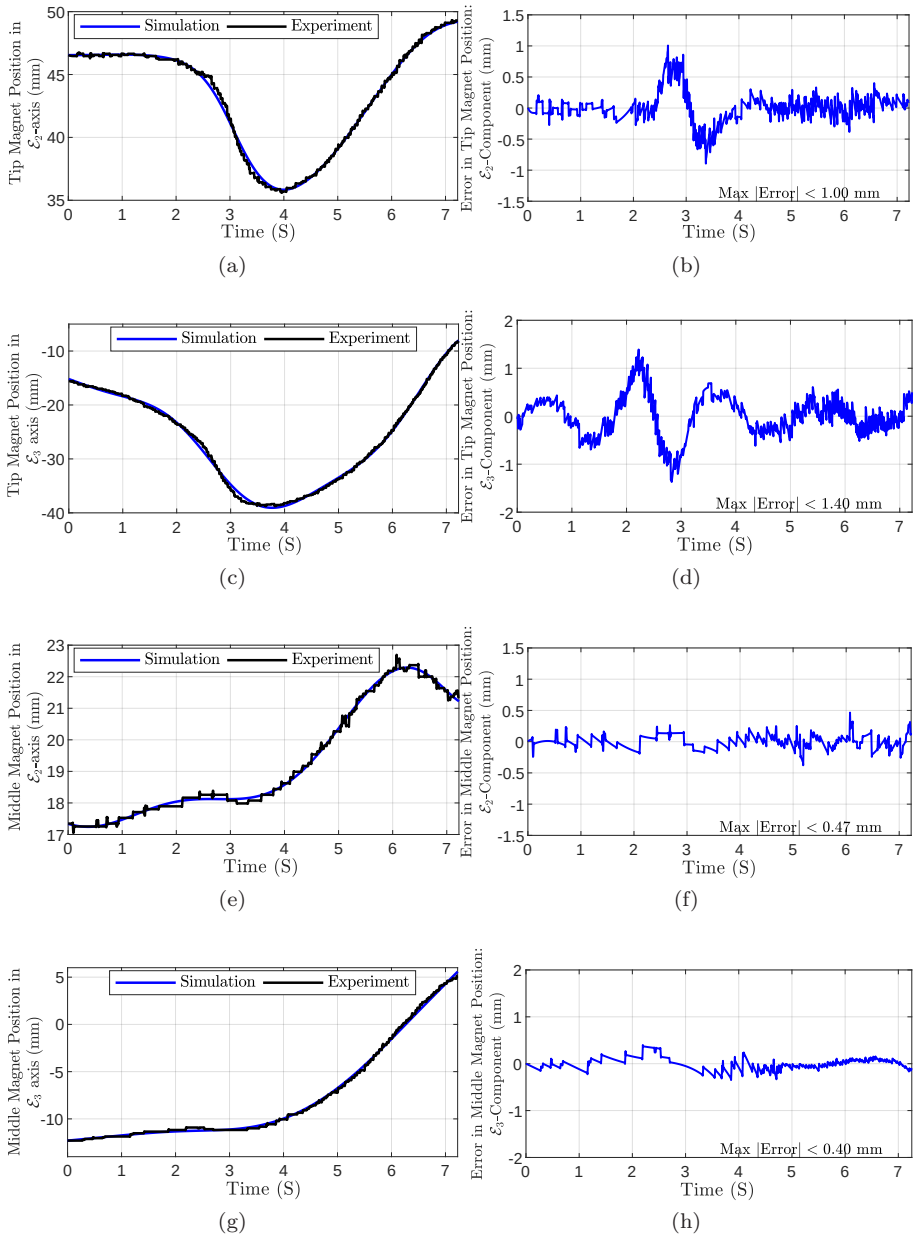


Figure I.15: Simulation and experiment results of Polydimethylsiloxane (PDMS) rod with two embedded magnets, 2D experiment: (a) Tip magnet's position in $\mathcal{E}_2$-axis direction. (b) Error of tip magnet's position in $\mathcal{E}_2$-axis direction. (c) Tip magnet's position in $\mathcal{E}_3$-axis direction. (d) Error of tip magnet's position in $\mathcal{E}_3$-axis direction. (e) Middle magnet's position in $\mathcal{E}_2$-axis direction. (f) Error of middle magnet's position in $\mathcal{E}_2$-axis direction. (g) Middle magnet's position in $\mathcal{E}_3$-axis direction. (h) Error of middle magnet's position in $\mathcal{E}_3$-axis direction.

Table I.8: Maximum and mean absolute errors in the experiments using flexible metal rods (AlSi05) and Polydimethylsiloxane (PDMS) rods.

Experiments		**Max. Error**	**Mean Absolute Error**
(1) Flexible rod (AlSi05): in-plane experiment		0.15 mm (i.e., 2.50%)	0.05 mm
(2) Flexible rod (AlSi05): out-of-plane experiment, both axes		0.15 mm (i.e., 2.12%)	0.05 mm
(3) PDMS rod (without magnet): in-plane experiment	$\mathcal{E}_2$-axis	0.56 mm (i.e., 4.87%)	0.05 mm
	$\mathcal{E}_3$-axis	0.28 mm (i.e., 4.89%)	0.05 mm
(4) PDMS rod (with magnet): in-plane experiment	$\mathcal{E}_2$-axis	0.61 mm (i.e., 3.35%)	0.14 mm
	$\mathcal{E}_3$-axis	0.55 mm (i.e., 1.16%)	0.13 mm
(5) PDMS rod (with magnet): circular motion		1.20 mm (i.e., 1.43%)	0.59 mm
(6) Square cross-section PDMS rod (with 2 magnets): in-plane experiment	Tip magnet: $\mathcal{E}_2$-axis	1.00 mm (i.e., 2.24%)	0.15 mm
	Tip magnet: $\mathcal{E}_3$-axis	1.40 mm (i.e., 5.13%)	0.33 mm
	Middle magnet: $\mathcal{E}_2$-axis	0.47 mm (i.e., 2.05%)	0.08 mm
	Middle magnet: $\mathcal{E}_3$-axis	0.40 mm (i.e., 3.68%)	0.10 mm

simulation is obtained as

$$\gamma_{corr} = \left(\frac{\omega_{nf_{\bar{N}}}}{\omega_{nf}}\right)^2$$

By employing this correction factor in the simulation, the effect of the number of discretization elements on the frequency of motion can be eliminated.

High fidelity models are helpful for explaining and predicting the behavior of a system with complex dynamics. However, due to computational constraints, these models may not be employed for closed-loop control purposes in a real-time implementation of robotic applications. Additionally, recent developments in computer simulations demand superior, robust, and efficient numerical frameworks compared to traditional approaches. Discrete geometric mechanics, which are employed in this paper, provides a systematic method to cope with the complexity of continuum manipulators' dynamics. The necessity of guaranteeing robots' performance in sensitive applications such as minimally invasive surgeries requires the use of pre-existing knowledge or a model in control architecture to obtain guaranteed and reliable behavior in the presence of disturbances and

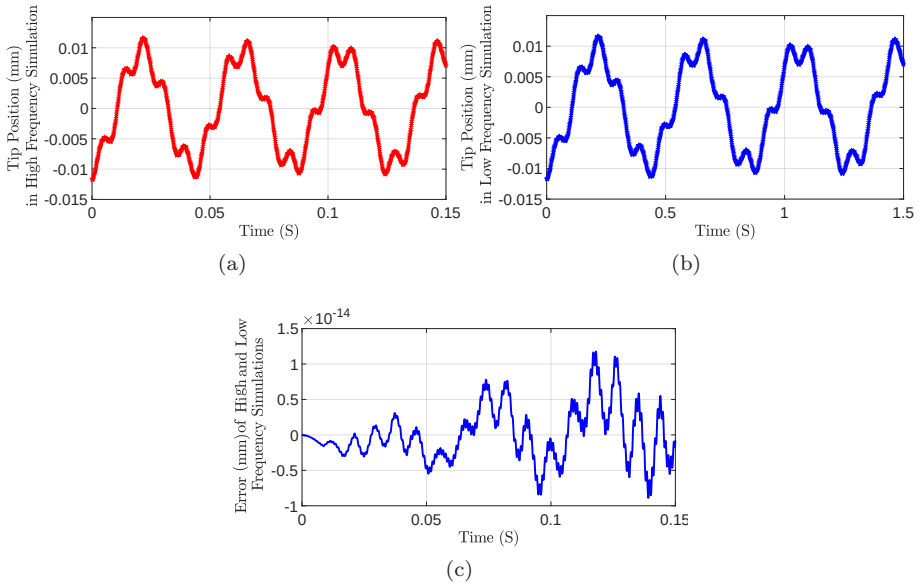


Figure I.16: Simulation results for high and low frequency comparison: (a) High frequency. (b) Low frequency. (c) Error of in high and low frequency cases.

uncertainties. Although model-free control approaches are easy to implement, they do not provide and ensure any performance level and high control-loop bandwidths.

I.6 Conclusions and Future Work

This article studies the estimation and model validation problem of continuum manipulators' dynamics using Lie group variational integrators. Using magnetic actuation, dynamic and static experiments were conducted on manipulators with rigid and soft materials (e.g., Aluminum and PDMS) to illustrate the validity of the presented algorithm for a wide range of experiments.

Due to the lack of knowledge about friction/damping, distributed predictive filters were designed to provide information about the unknown signals. Therefore, the dynamical model equipped with the estimation algorithm is a self-contained generic model for continuum manipulator integration, which provides us with a systematic approach to employ optimal control theory for realistic trajectory planning in the presence of user/environment-specified constraints. The designing of a controller and the parallel variational integration algorithm are to be investigated as future work.

Appendix I.A Preliminaries on Lie groups and Lie algebra

Some concepts and definitions on Lie groups and Lie algebra are presented (References [Dem+15] and [Hol11]).

Preliminaries on Lie groups: A group G is a set of elements with:

- A multiplication, $G \times G \to G$, such that the following properties hold:
  i. The product of g and h is written gh.
  ii. The product is associative, $(gh)k = g(hk)$.
- Identity element e: $eg = g$ and $ge = g$, $\forall g \in G$.
- Inverse operation $G \to G$, so that $gg^{-1} = g^{-1}g = e$.

A Lie group is a smooth manifold G which is also a group and for which the group operations of multiplication, $(g, h) \to gh$ for $g, h \in G$, and inversion, $g \to g^{-1}$ with $gg^{-1} = g^{-1}g = e$, are smooth functions. The action of a Lie group G on a manifold M is a group of transformations of M associated to elements of the group G, whose composition acting on M is corresponds to group multiplication in G. Left and right multiplication by $g \in G$ are denoted by $L_g, R_g : G \to G, L_g(f) = gf, R_g(f) = fg$. The tangent lifted actions $TL_g, TR_g : TG \to TG$ are sometimes denoted as $gv_f := TL_g(v_f)$ and $v_f g := TR_g(v_f)$ for simplicity, where $v_f \in TG$. Similarly, the cotangent lifted actions $T^*L_{g^{-1}}, T^*R_{g^{-1}} : T^*G \to T^*G$ is denoted by $g\alpha_f := T^*L_{g^{-1}}(\alpha_f)$, $\alpha_f g := T^*R_{g^{-1}}(\alpha_f)$, where $\alpha_f \in T^*G$.

Consider $G = SE(3)$, we identify the Lie algebra $\mathfrak{se}(3)$ of G with $\mathbb{R}^3 \times \mathbb{R}^3$ by using the hat map (I.2). Therefore, the adjoint action is written as:

$$\mathrm{Ad}_{(\mathcal{O}_{a_i}, \mathcal{P}_{a_i})}\Big(\mathcal{O}_{a_k}, \mathcal{P}_{a_k}\Big) = \Big(\mathcal{O}_{a_i}\mathcal{O}_{a_k}, \mathcal{O}_{a_i}\mathcal{P}_{a_k} + \mathcal{P}_{a_i} \times \mathcal{O}_{a_i}\mathcal{O}_{a_k}\Big)$$

also the coadjoint action reads

$$\mathrm{Ad}^*_{(\mathcal{O}_{a_i}, \mathcal{P}_{a_i})^{-1}}\Big(\mathcal{O}_{a_k}, \mathcal{P}_{a_k}\Big) = \Big(\mathcal{O}_{a_i}\mathcal{O}_{a_k} + \mathcal{O}_{P_i} \times \mathcal{O}_{a_i}\mathcal{P}_{a_k}, \mathcal{O}_{a_i}\mathcal{P}_{a_k}\Big)$$

The cotangent lift of left translation $T^*_e L_{(\mathcal{O}_{a_i}, \mathcal{P}_{a_i})} : T_{(\mathcal{O}_{a_i}, \mathcal{P}_{a_i})}G \to T^*_e G$ reads

$$T^*_e L_{(\mathcal{O}_{a_i}, \mathcal{P}_{a_i})}\Big(\mathcal{O}_{a_k}, \mathcal{P}_{a_k}\Big) = \Big(\mathcal{O}^T_{a_i}\mathcal{O}_{a_k}, \mathcal{O}^T_{a_i}\mathcal{P}_{a_k}\Big)$$

where $e = (I, 0)$ is the identity element.

References

[AM78] Abraham, R. and Marsden, J. E. *Foundations of mechanics*. 2d ed., rev., enl., and reset. Reading, Mass: Benjamin/Cummings Pub. Co, 1978.

[BS99] Bobenko, A. I. and Suris, Y. B. "Discrete Time Lagrangian Mechanics on Lie Groups with an Application to the Lagrange Top". In: *Communications in mathematical physics* vol. 204, no. 1 (1999), pp. 147–188.

[Bur+13] Burgner, J. et al. "Debulking from within: a robotic steerable cannula for intracerebral hemorrhage evacuation". In: *IEEE Transactions on Biomedical Engineering* vol. 60, no. 9 (2013), pp. 2567–2575.

[Cam+08] Camarillo, D. B. et al. "Mechanics modeling of tendon-driven continuum manipulators". In: *IEEE Transactions on Robotics* vol. 24, no. 6 (Dec. 2008), pp. 1262–1273.

[CCS09] Camarillo, D., Carlson, C., and Salisbury, J. "Configuration tracking for continuum manipulators with coupled tendon drive". In: *IEEE Transactions on Robotics* vol. 25, no. 4 (Aug. 2009), pp. 798–808.

[Cia+13] Cianchetti, M. et al. "STIFF-FLOP surgical manipulator: Mechanical design and experimental characterization of the single module". In: *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2013, pp. 3576–3581.

[CP09] Carpi, F. and Pappone, C. "Stereotaxis Niobe® magnetic navigation system for endocardial catheter ablation and gastrointestinal capsule endoscopy". In: *Expert review of medical devices* vol. 6, no. 5 (2009), pp. 487–498.

[Dem+15] Demoures, F. et al. "Discrete variational Lie group formulation of geometrically exact beam dynamics". en. In: *Numerische Mathematik* vol. 130, no. 1 (May 2015), pp. 73–123.

[GC15] Greigarn, T. and Cavusoglu, M. C. "Pseudo-rigid-body model and kinematic analysis of MRI-actuated catheters". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. Seattle, WA, USA: IEEE, May 2015, pp. 2236–2243.

[GDS19] Grazioso, S., Di Gironimo, G., and Siciliano, B. "A geometrically exact model for soft continuum robots: The finite element deformation space formulation". In: *Soft robotics* vol. 6, no. 6 (2019), pp. 790–811.

[Gra+00] Grady, M. S. et al. "Experimental study of the magnetic stereotaxis system for catheter manipulation within the brain". In: *Journal of neurosurgery* vol. 93, no. 2 (2000), pp. 282–288.

[Guo+96] Guo, S. et al. "Micro active guide wire catheter system-Characteristic evaluation, electrical model and operability evaluation of micro active catheter". In: *Proceedings of IEEE International Conference on Robotics and Automation*. Vol. 3. Minneapolis, MN, USA: IEEE, 1996, pp. 2226–2231.

[Hol11] Holm, D. D. *Geometric Mechanics: Part II: Rotating, Translating and Rolling*. en. 2nd ed. IMPERIAL COLLEGE PRESS, Oct. 2011.

[HSM18] Heunis, C., Sikorski, J., and Misra, S. “Flexible instruments for endovascular interventions: Improved magnetic steering, actuation, and image-guided surgical instruments”. In: *IEEE robotics & automation magazine* vol. 25, no. 3 (2018), pp. 71–82.

[JGT09] Jones, B. A., Gray, R. L., and Turlapati, K. “Three dimensional statics for continuum robotics”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. ISSN: 2153-0866. Oct. 2009, pp. 2659–2664.

[JPZ14] Jung, J., Penning, R. S., and Zinn, M. R. “A modeling approach for robotic catheters: effects of nonlinear internal device friction”. en. In: *Advanced Robotics* vol. 28, no. 8 (Apr. 2014), pp. 557–572.

[KH11] Kesner, S. B. and Howe, R. D. “Position control of motion compensation cardiac catheters”. In: *IEEE Transactions on Robotics* vol. 27, no. 6 (2011), pp. 1045–1055.

[KS10] Kai Xu and Simaan, N. “Intrinsic wrench estimation and its performance index for multisegment continuum robots”. In: *IEEE Transactions on Robotics* vol. 26, no. 3 (June 2010), pp. 555–561.

[Lee08] Lee, T. *Computational geometric mechanics and control of rigid bodies*. 2008.

[LMC16] Laschi, C., Mazzolai, B., and Cianchetti, M. “Soft robotics: Technologies and systems pushing the boundaries of robot abilities”. en. In: *Science Robotics* vol. 1, no. 1 (Dec. 2016), eaah3690.

[Loc+10] Lock, J. et al. “Quasistatic modeling of concentric tube robots with external loads”. eng. In: *Proceedings of the ... IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE/RSJ International Conference on Intelligent Robots and Systems* vol. 2010 (Dec. 2010), pp. 2325–2332.

[LOL14] Leitz, T., Ober-Blöbaum, S., and Leyendecker, S. “Variational Lie Group Formulation of Geometrically Exact Beam Dynamics: Synchronous and Asynchronous Integration”. en. In: *Multibody Dynamics: Computational Methods and Applications*. Ed. by Terze, Z. Computational Methods in Applied Sciences. Cham: Springer International Publishing, 2014, pp. 175–203.

[Lu95] Lu, P. “Optimal predictive control of continuous nonlinear systems”. In: *International Journal of Control* vol. 62, no. 3 (1995), pp. 633–649.

[MD11] Mahvash, M. and Dupont, P. E. “Stiffness control of surgical continuum manipulators”. In: *IEEE Transactions on Robotics* vol. 27, no. 2 (2011), pp. 334–345.

[MR94] Marsden, J. E. and Ratiu, T. S. *Introduction to mechanics and symmetry*. Vol. 17. Texts in Applied Mathematics. New York, NY: Springer New York, 1994.

[Pai02] Pai, D. K. “Strands: interactive simulation of thin solids using cosserat models”. en. In: *Computer Graphics Forum* vol. 21, no. 3 (Sept. 2002), pp. 347–352.

[Pol+17] Polygerinos, P. et al. “Soft robotics: review of fluid-driven intrinsically soft devices; manufacturing, sensing, control, and applications in human-robot interaction: review of fluid-driven intrinsically soft robots”. en. In: *Advanced Engineering Materials* vol. 19, no. 12 (Dec. 2017), p. 1700016.

[RJW10] Rucker, D. C., Jones, B. A., and Webster III, R. J. “A geometrically exact model for externally loaded concentric-tube continuum robots”. In: *IEEE transactions on robotics* vol. 26, no. 5 (2010), pp. 769–780.

[RM16] Roesthuis, R. J. and Misra, S. “Steering of multisegment continuum manipulators using rigid-link modeling and fbg-based shape sensing”. In: *IEEE Transactions on Robotics* vol. 32, no. 2 (Apr. 2016), pp. 372–382.

[RT15] Rus, D. and Tolley, M. T. “Design, fabrication and control of soft robots”. en. In: *Nature* vol. 521, no. 7553 (May 2015), pp. 467–475.

[Ruc+10] Rucker, D. C. et al. “Equilibrium conformations of concentric-tube continuum robots”. en. In: *The International Journal of Robotics Research* vol. 29, no. 10 (Sept. 2010), pp. 1263–1280.

[Sik+17] Sikorski, J. et al. “Introducing BigMag — A novel system for 3D magnetic actuation of flexible surgical manipulators”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. Singapore, Singapore: IEEE, May 2017, pp. 3594–3599.

[Sik+19] Sikorski, J. et al. “The ARMM system: An optimized mobile electromagnetic coil for non-linear actuation of flexible surgical instruments”. In: *IEEE transactions on magnetics* vol. 55, no. 9 (2019), pp. 1–9.

[Sim85] Simo, J. C. “A finite strain beam formulation. The three-dimensional dynamic problem. Part I”. In: *Computer methods in applied mechanics and engineering* vol. 49, no. 1 (1985), pp. 55–70.

[Tho+20] Thomas, T. et al. “A Monolithic Compliant Continuum Manipulator: a Proof-of-Concept Study”. en. In: *Journal of Mechanisms and Robotics* (Apr. 2020), pp. 1–11.

[Tun11] Tunay, I. “Distributed parameter statics of magnetic catheters”. eng. In: *Conference proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference* vol. 2011 (2011), pp. 8344–8347.

[Tun13] Tunay, I. “Spatial continuum models of rods undergoing large deformation and inflation”. In: *IEEE Transactions on Robotics* vol. 29, no. 2 (Apr. 2013), pp. 297–307.

[VSM19] Venkiteswaran, V. K., Sikorski, J., and Misra, S. “Shape and contact force estimation of continuum manipulators using pseudo rigid body models”. en. In: *Mechanism and Machine Theory* vol. 139 (Sept. 2019), pp. 34–45.

[Wen+12] Wen Tang et al. “A stable and real-time nonlinear elastic approach to simulating guidewire and catheter insertions based on cosserat rod”. In: *IEEE Transactions on Biomedical Engineering* vol. 59, no. 8 (Aug. 2012), pp. 2211–2218.

[WJ10] Webster III, R. J. and Jones, B. A. “Design and kinematic modeling of constant curvature continuum robots: A review”. In: *The International Journal of Robotics Research* vol. 29, no. 13 (2010), pp. 1661–1683.

[WRC09] Webster, R., Romano, J., and Cowan, N. “Mechanics of precurved-tube continuum robots”. In: *IEEE Transactions on Robotics* vol. 25, no. 1 (Feb. 2009), pp. 67–78.

[XS08] Xu, K. and Simaan, N. “An investigation of the intrinsic force sensing capabilities of continuum robots”. In: *IEEE Transactions on Robotics* vol. 24, no. 3 (June 2008), pp. 576–587.

[XS10] Xu, K. and Simaan, N. “Analytic formulation for kinematics, statics, and shape restoration of multibackbone continuum robots via elliptic integrals”. en. In: *Journal of Mechanisms and Robotics* vol. 2, no. 1 (Feb. 2010), p. 011006.

Authors’ addresses

Abbas Tariverdi Department of Physics, University of Oslo, Sem Sælands 24, 0371 Oslo, Norway, abbast@uio.no

Venkatasubramanian K. Venkiteswaran Department of Biomechanical Engineering, University of Twente, 7500 AE Enschede, The Netherlands, v.kalpathyvenkiteswaran@utwente.nl

Ørjan G. Martinsen Department of Physics, University of Oslo, Sem Sælands 24, 0371 Oslo, Norway, o.g.martinsen@fys.uio.no

Ole J. Elle The Intervention Centre, Oslo University Hospital, Oslo, Norway, oelle@ous-hf.no

Jim Tørresen Department of Informatics, University of Oslo, Gaustadalleen 23B, Oslo, Norway, jimtoer@ifi.uio.no

Sarthak Misra Department of Biomechanical Engineering, University of Twente, 7500 AE Enschede, The Netherlands, s.misra@utwente.nl

Paper II

A Recurrent Neural-Network-Based Real-Time Dynamic Model for Soft Continuum Manipulators

II

Abbas Tariverdi[1], **Venkatasubramanian K. Venkiteswaran, Michiel Richter, Ole J. Elle, Jim Tørresen, Kim Mathiassen, Sarthak Misra, Ørjan G. Martinsen**

Published in *Frontiers in Robotics and AI*, March 2021, volume 8, pp. 631303.
DOI: 10.3389/frobt.2021.631303.

Abstract

This paper introduces and validates a real-time dynamic predictive model based on a neural network approach for soft continuum manipulators. The presented model provides a real-time prediction framework using neural-network-based strategies and continuum mechanics principles. A time-space integration scheme is employed to discretize the continuous dynamics and decouple the dynamic equations for translation and rotation for each node of a soft continuum manipulator. Then the resulting architecture is used to develop distributed prediction algorithms using recurrent neural networks. The proposed RNN-based parallel predictive scheme does not rely on computationally intensive algorithms; therefore, it is useful in real-time applications. Furthermore, simulations are shown to illustrate the approach performance on soft continuum elastica, and the approach is also validated through an experiment on a magnetically-actuated soft continuum manipulator. The results demonstrate that the presented model can outperform classical modeling approaches such as the Cosserat rod model while also shows possibilities for being used in practice.

The paper was partially supported by the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation programme (Grant Agreement #866494 - project MAESTRO).

[1]University of Oslo, Postboks 1337 Blindern, 0316 Oslo, Norway, abbast@uio.no

Contents

II.1	Introduction	118
II.2	Problem Statement	120
II.3	Proposed RNN-based Model	121
II.4	Simulations Results	122
II.5	Experimental Results	131
II.6	Discussion	142
II.7	Conclusion	143
II.A	Cosserat Rod Theory	143
References		144

II.1 Introduction

Soft continuum manipulators are flexible and highly deformable robots composed of soft and mostly elastic materials, and can serve as possible substitutes for rigid robots. Advantages of soft manipulator robots such as their compliance, dexterity, and adaptability to complex workspaces are driving the emergent research in this field. By contrast, rigidity of traditional rigid robots limits their use in constrained and confined environments, and reduces the possibilities for safe interaction with humans. Soft continuum manipulators have found applications in many areas, such as dexterous grasping [KMR15; McM+06] and assistive devices [Ans+17], and particularly in the field of minimally invasive surgeries, such as laryngeal surgery [STF04], catheter-based endovascular intervention [Bur+13; Gra+00], and cardiovascular surgery [KH11].

Analytical modeling of soft manipulators helps evaluate their motion and determine their workspace, in order to be used for control, motion planning, and animation purposes. Soft manipulators distinguish themselves by having an infinite number of degrees of freedom in any workspace they occupy. This characterization makes modeling complicated for soft manipulators. Several approaches have been investigated thus far in the literature. Most of the approaches consider the kinematic (i.e. static or quasi-static) modeling of the manipulators such as static analysis using virtual-work model [KS10], Cosserat rod theory [JGT09; MD11; Pai02], and α Lie group formulation [GDS19]. These models do not describe full dynamics of the manipulators, and they may show performance degradation when it comes to high-frequency applications or large and complex deformations. On the other hand, dynamical modeling approaches (e.g. [HWK19; JPZ14; Sad+19; Tar+20; TAR19; Wen+12]), contain dynamics of the manipulators and also take into account time-varying responses of manipulators, including high-frequency modes. However, the dynamic models mostly rely on traditional methods, such as finite elements and finite differences (i.e., quantitative and numerical methods), making the algorithms computationally expensive for real-time applications. In other words, to obtain sufficiently accurate solutions, methods need to deal with fine meshes, which increase memory use and computation time. Another limitation is that their

solutions are discrete or not sufficiently differentiable. It is worth noting that in model-based controllers or observers, having a differentiable solution (i.e., a solution that can be evaluated continuously on the workspace) is crucial in the design process. Furthermore, when softer materials are employed for manipulator construction with more complex geometries or large deformations, modeling their behavior analytically becomes challenging. Therefore, there is a need for appropriate data-driven approaches without compromising computational bandwidths and the prediction quality.

Dynamics of soft continuum manipulators have highly nonlinear behavior and are expressed as Partial Differential Equations (PDEs). An effective approach to represent and model PDEs solutions is to use Neural Networks (NN). NN-based solutions of PDEs are infinitely differentiable by eliminating the need for interpolation. Furthermore, compared to finite elements or difference methods, solutions are represented by fewer parameters, which reduces the memory use. There are studies that use machine learning algorithms to find a solution for special types of PDEs such as [EHJ17; LK90; LLF98; RPK19]. However, to the authors' best knowledge, there is no study that investigates possible NN-based solutions for partial differential equations that describe the full dynamics of continuum manipulators. In this work, inspired by a time-space integration scheme and by using the Lie group variational integration method [Dem+15], the dynamic equations for translation and rotation for each node of a soft continuum manipulator are decoupled, providing an appropriate structure aimed at developing a real-time modeling algorithm. Afterward, Recurrent Neural Networks (RNNs)-based models are employed to approximate the high-dimensional discretized equations. Additionally, external torques and forces (e.g., control inputs, friction, and gravity) are incorporated into the model in a real-time manner for control applications.

The ability of RNNs to learn and approximate large classes of nonlinear functions over sequences of inputs accurately makes them prime candidates for use in dynamic modeling of complex nonlinear systems. RNNs with Long Short-Term Memory (LSTM) layers process sequences by iterating through the sequence elements. Using an internal feedback, the network is capable of preserving long-term dependencies. Essentially, LSTM layers prevent older information from gradually vanishing. These networks also have been used for several applications in soft robotics. To name a few, Thuruthel in [Thu+19] proposes a model-free, real-time sensing method for soft robots perception. The authors in [Thu+17] uses RNNs to model and control soft robotic manipulators. Also, force and motion estimation using RNNs has been investigated in [Mar+19] and [Tur+18], respectively.

This paper aims to develop a real-time dynamic model for analyzing the dynamics of soft manipulators. Investigation of previous work on the modeling of the continuum manipulators suggests that existing literature focuses primarily on static or quasi-static approaches, or does not provide a real-time model. The contribution of this article is to present a scalable, parallel and real-time modeling algorithm for soft manipulators dynamics. The contributions of this paper are as follows.

- Existing approaches primarily deal with kinematic modeling methods. Nevertheless, in this study, real-time prediction of soft manipulators full spatial dynamics is considered in the proposed RNN-based algorithm by proposing multiple light-weight RNN-based models.

- In traditional modeling approaches, there are no systematic methods to obtain knowledge about dissipation forces, in particular friction, in the modeling procedure. The presented algorithm intrinsically takes the dissipation forces into account and incorporates their effects into the model.

- Through an experiment, results of the proposed RNN-based model and Cosserat rod theory method are compared, revealing the practical effectiveness of the proposed methodology.

The remainder of this paper is organized as follows: the problem statement is given in Section II.2. Section II.3 describes the proposed RNN-based algorithm in details. In Sections II.4 and II.5, different simulations and experimental validation are presented to demonstrate the efficacy of the proposed RNN-based method, in terms of the model performances and accurately predicting poses of manipulators. Finally, the discussion and main conclusions are stated in Sections II.6 and II.7.

II.2 Problem Statement

Consider a continuum manipulator with large deflections described by dynamic equations of motion (as presented in [Tar+20] and [Dem+15]) in the PDEs form as

$$\begin{aligned} \boldsymbol{H}\boldsymbol{\omega}_t + \boldsymbol{\omega} \times \boldsymbol{H}\boldsymbol{\omega} + \boldsymbol{n} \times \boldsymbol{\Lambda}^{-1}\boldsymbol{\phi}_x - \boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x \times \boldsymbol{m} - \boldsymbol{m}_x &= \boldsymbol{\Lambda}^{-1}\boldsymbol{\tau} \\ M\boldsymbol{\phi}_{tt} - \boldsymbol{\Lambda}(\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_x \times \boldsymbol{n}) - \boldsymbol{\Lambda}\boldsymbol{n}_x + \boldsymbol{f}^c &= \boldsymbol{f} \end{aligned} \tag{II.1}$$

where $M = \rho \times A$ (ρ and A are the manipulator constant mass density and its cross-section area), $\boldsymbol{\omega} \in \mathbb{R}^3$ is the manipulator's angular velocity, $\boldsymbol{H} \in \mathbb{R}^{3\times3}$ is the manipulator's inertia matrix, $\boldsymbol{\phi} \in \mathbb{R}^3$ is the position of the manipulator's line of centroids in its workspace, $\boldsymbol{\Lambda} \in SO(3)$ denotes the orientation of moving cross-sections at point ϕ. Also, $\boldsymbol{n} \in \mathbb{R}^3$ and $\boldsymbol{m} \in \mathbb{R}^3$ are the stresses and momenta along the manipulator, $\boldsymbol{f}^c \in \mathbb{R}^3$ represents conservative forces (e.g. gravity). Furthermore, $(\cdot)_x$, $(\cdot)_t$, and $(\cdot)_{tt}$ denote partial derivatives with respect to position, time, and the second partial derivative with respect to time, respectively. Finally, $\boldsymbol{f} \in \mathbb{R}^3$ and $\boldsymbol{\tau} \in \mathbb{R}^3$ are non-conservative forces and torques (e.g., frictions and control inputs) [2].

Although high fidelity models given in the references can describe continuum manipulators dynamics efficiently, they suffer from limitations that are discussed in Section II.1. Inspired by the structure and formulation of the dynamics based

[2] For the details see [Dem+15].

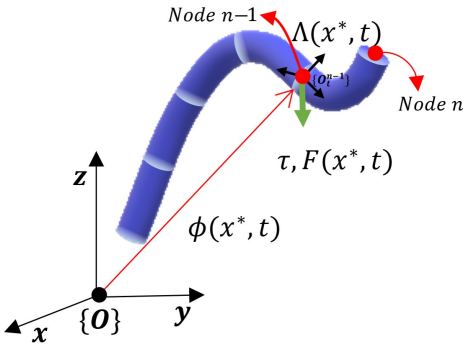


Figure II.1: A soft manipulator at time t with discretization nodes n and n–1 are shown. $\phi(x^*,t)$ and $\Lambda(x^*,t)$ denote the position and the orientation of cross-section of Node n–1, respectively. In addition, the force $F(x^*,t)$, torque $\tau(x^*,t)$, and the conservative force f^c (e.g., gravity) are applied to Node n–1 at the position $\phi(x^*,t)$.

on the Lie group variational integration scheme, the aim is to propose distributed deep recurrent neural networks to capture and simulate soft manipulators dynamics in real-time to be able control them more accurately than existing models.

II.3 Proposed RNN-based Model

This section is devoted to develop a model based on the time series prediction using RNNs. To solve PDEs numerically using NNs, one approach is to utilize discrete solutions of finite element or difference methods to train an NN. A Lie group variational time integration model is employed to discretize the continuous dynamics of a soft manipulator [3]. The whole manipulator is discretized into an arbitrary number of nodes where the position and orientation equations of each node are decoupled. In our study, we discretize the manipulator with equidistant nodes, but this can be changed depending on the application.

Figure II.1 demonstrates a soft continuum manipulator at time t where x^* is the undeformed length of Node n–1. The force $F(x^*,t)$, torque $\tau(x^*,t)$ are applied to Node n–1 at the position $\phi(x^*,t)$. Also, $\Lambda(x^*,t)$ is the orientation matrix from the frame $\{O\}$ to the frame $\{O_t^{n-1}\}$ attached to the cross-section of Node n–1.

The discrete equations suggest an appropriate structure for the RNNs-based model.

Given time-sequence inputs (as a first input layer), i.e., poses (positions and orientations) of nodes, and also forces and torques (as a second input layer) applied to each node, the RNN-based model of Node n is depicted in

[3] Dynamic equations are given in [Tar+20, Sec. 2]

Figure II.2:(A,B). For Node n, the first input layer is a time-sequence series of poses $\boldsymbol{p_{n-1}}$, $\boldsymbol{p_n}$, and $\boldsymbol{p_{n+1}}$ (i.e., poses of Node n and its adjacent nodes n–1 and n+1) and the second input layer includes forces and torques of node n at time t, i.e., $[\boldsymbol{F_t^n}, \boldsymbol{\tau_t^n}]^T$ which are incorporated into the model through dense layers.

The network takes specific size vectors as inputs, which are called input layers. The inputs are transformed through a series of hidden layers (LSTM, dense, or fully connected layers) to produce an output. The output vectors are called an output layer. Dense or fully connected layers perform linear operations (i.e., multiplication and summation) on their inputs. Furthermore, LSTM layers consist of LSTM units, which can process sequences of data of any length, for example, poses of nodes. An LSTM unit controls contributions of each element of the input layer in the output and keeps track of the dependencies between the elements [HS97].

For the training process, data-sets contain time-sequence inputs and forces and torques applied to each node. Also, for each node, the poses of the node and its neighbors are considered features, as shown in Figure II.2:(A,B). The first and second input layers proceed through LSTM layers and dense layers as hidden layers, respectively. Finally, output layers have resulted from fully connected layers.

By augmenting the given models for all nodes (see Figure II.2:(B)) as a series, the proposed RNN-based models of the whole continuum manipulator with N nodes with non-conservative forces and torques are depicted in Figure II.2:(C). Output of every node is updated at each time step by using a history (at least two previous time steps) of neighboring node outputs. Therefore, the proposed architecture suggests a suitable framework to construct a parallel modeling algorithm.

II.4 Simulations Results

In this section, we consider different examples and evaluate the performance of the proposed RNN-based model in Figure II.2:(C). It is worth mentioning that data-sets play a crucial role in efficiency and accuracy in machine learning-based algorithms. The data acquisition process from a robot in real-world environments is both time and cost-consuming (implementation of multiple sensors, data filtering, and fusion, etc.). As an alternative approach, the required data can be acquired through simulations of high fidelity models. The obtained data can thus be transferred to train the algorithms to be implemented in real-world scenarios. In this section and for the presented examples, required data for the training of the proposed RNN-based model are acquired through simulations of the algorithm presented in [Tar+20, Sec. 2]. For clarity, this model is henceforth referred to as the analytical dynamic model. In addition, since thin rods are considered in the examples, orientations of cross-sections are not of any concern. Also, it should be noted that orientations, except the twisting angle, can be reconstructed from manipulators' configuration. Therefore, to obtain a computationally-light model, the focus of attention is only on the

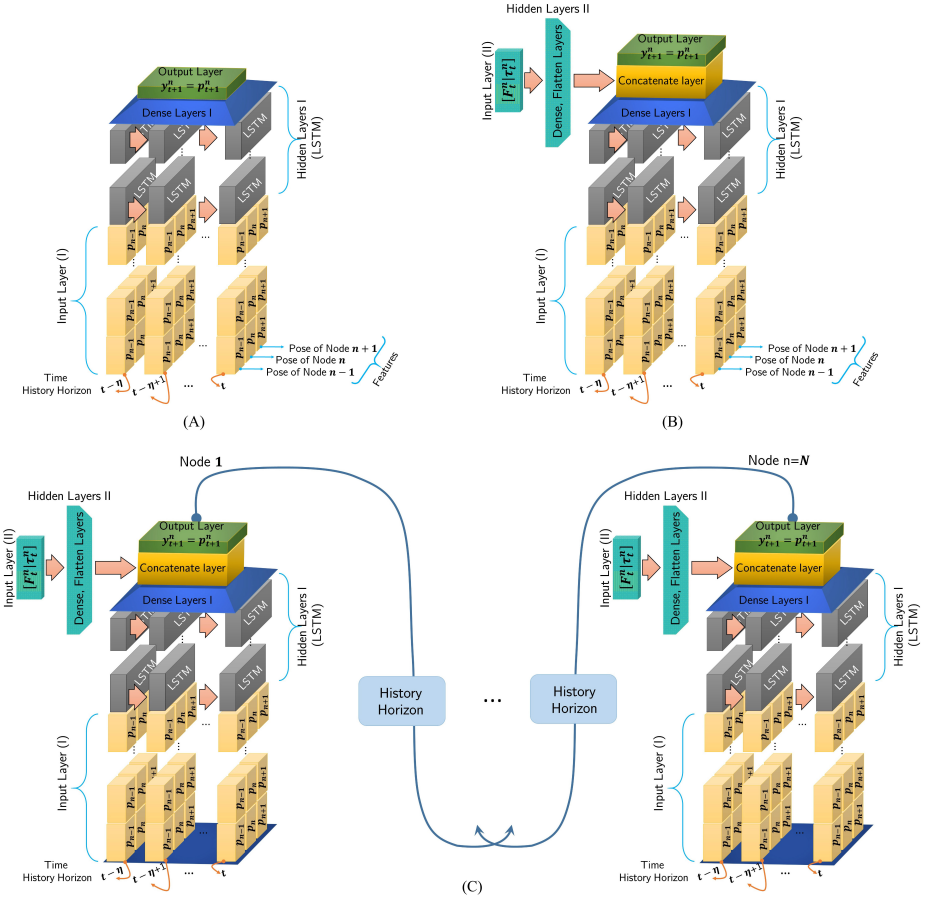


Figure II.2: Recurrent Neural Network-based model, length of the time history horizon is determined by η and features composed of adjacent nodes pose: **(A)**: Poses of Nodes n–1, n, and n+1 are the input layer and no forces or torques are applied to the node. **(B)**: The first input layer is composed of poses $\boldsymbol{p_{n-1}}$, $\boldsymbol{p_n}$, and $\boldsymbol{p_{n+1}}$ at time history horizon $[t–\eta, t–\eta+1, \cdots, t]$ and the second input layer includes forces and torques $[\boldsymbol{F_t^n}, \boldsymbol{\tau_t^n}]^T$ which are incorporated into the model through the Hidden Layers II (dense and flatten layers). **(C)**: Proposed RNN-based models of the continuum manipulator with N nodes including Input, Hidden, and Output Layers. A history of each node output is used as an input for adjacent nodes. Nodes poses (Input Layer I) and forces and torques (Input Layer II) through the Hidden layers I and II are proceed and concatenated together.

prediction of positions.

II.4.1 First Simulation: An Ellipse without External Wrenches

As a first case, a cylindrical rod is bent into a circle and its ends are attached to one another. The rod is then deformed into an elliptical shape and released. Due to potential energies in the ellipse, it starts to move without any external disturbances. The goal is to model the behavior of the ellipse resultant from its internal elastic energy.

The ellipse is formed in the $\boldsymbol{xy}$-plane with the width 0.2 m and height 0.6 m. As boundary conditions, the first and last nodes are fixed to the origin and their orientations are set to $\mathtt{R}_{\mathtt{y}}(18.07°)$ and $\mathtt{R}_{\mathtt{y}}(341.92°)$, respectively, where $\mathtt{R}_{\mathtt{y}}(\theta)$ denotes a rotation matrix describing a rotation around the y-axis by θ degrees. The rod properties, simulation parameters, and the structure of the proposed RNN-based model are given in Figure II.3:(A,B). Furthermore, the initial and a few time-evolved configurations are shown in Figure II.3:(C). As seen, the ellipse oscillates back and forth due to its internal elastic energy. Since orientations except the twisting angle can be reconstructed from the configuration of the manipulator, to have a light model and for brevity, positions of the node located at $(-0.01, -0.59)$ — Node 30^{th} — are predicted. The chosen node is the furthest from the origin and would, compared to other nodes, most likely have the largest errors.

50001 position samples are generated from the analytical model for each node. We augment 1-by-3 position vectors of Node 30^{th} and its adjacent nodes (Nodes 29^{th} and 31^{st}) at each time step. Therefore, the augmentation results in a 1-by-9 vector. Furthermore, the size of history horizons is chosen to be 2. In other words, $\eta = 1$ in Figure II.2:(A, B). Finally, augmented 2-by-9 tensors are obtained for each time step. The prepared data-set is called Data-set I and 60 percent of it is used for training process. The architecture in Figure II.3:(A) shows the input layer consists of tensors of size 2×9. The first dimension of all layers are reserved for batch sizes and for the training, the batch size 1 was chosen. In the architecture of the model in Figure II.3:(A), the Input, Hidden and Output Layers I together with the number of nodes and type of layers are demonstrated according to Figure II.2:(A).

First, we evaluate the model by using unseen data samples in Data-set I and the results are shown in Figure II.3:(D). The maximum and mean absolute error are $(1.57 \text{ mm}, 0.27 \text{ mm})$, $(2.27 \text{ mm}, 0.46 \text{ mm})$, and $(0.23 \text{ mm}, 0.06 \text{ mm})$, or in other words, the percentage of the maximum errors with respect to the length of the manipulator are 0.11, 0.17, and 0.02 in the $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$-axes, respectively. It is worth mentioning that it is prior knowledge that the manipulator does not have any motion in the $\boldsymbol{z}$-axis and therefore, components of the $\boldsymbol{z}$-axis in position vectors can be ignored. Furthermore, the evaluation Root-Mean-Square (RMS) errors of the considered node in all axes at every time step is calculated

by

$$RMSE(t) = \sqrt{\frac{1}{3}\Big(\big(x_p(t) - x_m(t)\big)^2 + \big(y_p(t) - y_m(t)\big)^2 + \big(z_p(t) - z_m(t)\big)^2\Big)} \tag{II.2}$$

where predicted positions $\big[x_p(t), y_p(t), z_p(t)\big]^T$ obtained from the proposed RNN-based model and measurement positions $\big[x_m(t), y_m(t), z_m(t)\big]^T$ in Data-set I and the results are shown in Figure II.3:(E).

To demonstrate that the model can be extended to different boundary and initial conditions, the cylindrical rod is employed to form a horizontal ellipse with the width 0.6 m and height 0.2 m. The rod properties and the simulation parameters given in Figure II.3:(B) are used. As boundary conditions, the first and last nodes are attached to the origin and their orientations are set to the identity. The manipulator with the new boundary and initial conditions is only used for the evaluation of the trained model by predicting positions of the node located at $(-0.01, -0.19)$. Based on the prediction, the maximum and mean absolute errors are (26.33 mm, 3.37 mm), (21.71 mm, 3.70 mm), and (4.92 mm, 4.22 mm) in the $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$-axes, respectively. Furthermore, the maximum/worst-case errors with respect to the length of the manipulator are 1.97%, 1.62%, and 0.37% in each axis, respectively.

Let us assume that the analytical dynamic model is implemented in a parallel scheme, i.e., each node of 59 nodes is handled with a CPU core or different hardware such that there is no latency in communications. Then, the dynamics of each node can be solved in 1.62×10^{-4} s on average. In addition, to preserve the convergence of the solver of the analytical dynamic model, the maximum constant time step for this simulation is 10^{-3} s. A minimum criterion to have a real-time performance is that the time required to solve each node dynamics must be less than the constant step simulation. To be more specific, to have a real-time model, the CPU time, i.e., the amount of time spent in a user code must be less than Wall-clock time that measures the time elapsed to run a user code. According to this minimal criterion, as long as the computation-time for simulation of a method/model is less than a user-defined time for the simulation, the model is called a real-time model. It can be shown that in this example and based on the given assumption, the maximum bandwidth for a real-time performance is 3.93 Hz on average (calculation is done on a 16 GB, 1.99 GHz Intel i7 machine running windows 10). It should be pointed out that we use the same machine for calculations in this paper. It will be discussed that even achieving this bandwidth limit is not feasible. On the other hand, for the proposed RNN-based model, the bandwidth of a real-time performance is 65.70 Hz, which can be further improved by optimizing the number of layers and trainable parameters.

It is worth mentioning that the considered assumption is very strict, which cannot be satisfied in reality. First of all, conventional algorithms need a relatively high number of nodes to have numerical stability and an acceptable convergence rate. Furthermore, due to limitations in computation resources, more than

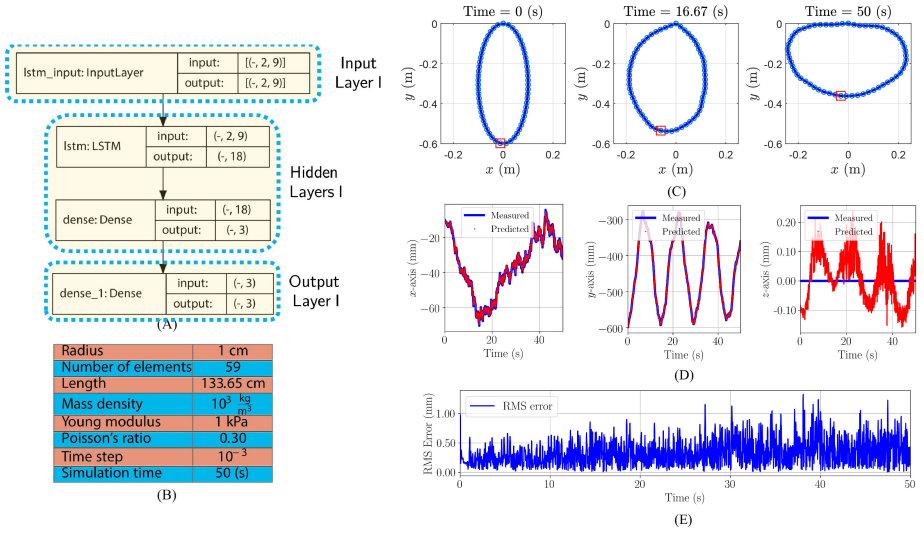


Figure II.3: First simulation example: **(A)**: The model architecture used for the first simulation example. The first stage is the Input layer, the intermediate stages are the Hidden layers, and the last stage is the Output layer. The first dimension of inputs and outputs in each layer are unspecified and can vary with the size of batches. **(B)**: Rod properties and simulation parameters used in the first example. **(C)**: Initial and time-evolved configurations. Positions of node specified by a red rectangle are measured and predicted. **(D)**: Measured position calculated from analytical dynamic model and predicted by the proposed RNN-based model in the $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$-axes are given. **(E)**: RMS error considering all the axes.

one node will be assigned to each core of CPU, and there is always latency in communications between threads in parallel programmings. Therefore, reaching the mentioned bandwidth through the analytical dynamic model is infeasible. However, the real-time performance of the proposed RNN-based model can be applicable in closed-loop control applications.

II.4.2 Second Simulation: A Cylindrical Rod with External Wrenches

In the second example, we simulate a rod with a circular cross section, which is actuated by external forces such that its tip tracks a square in space. In this example, the goal is to model the behavior of the rod which results from applied external forces on its end-effector. For boundary conditions, the first node is fixed to the origin and its orientation is set to the identity for all time steps. The rod properties and simulation parameters, and the structure of the proposed model are given in Figure II.4. The trajectory of the end-effector and the applied forces onto it are shown in Figure II.5:(A).

160001 position and force samples are generated from the analytical model for each node. We augment 1-by-3 position vectors of the last node (end-effector) and its adjacent node at each time step. Therefore, the augmentation results in a 1-by-6 vector. Furthermore, the size of history horizons is chosen to be 2 ($\eta = 1$). Finally, augmented 2-by-6 tensors are obtained for each time steps which are fed to the model as the Input Layer I. The same preparation process are applied for the force data samples which are used as the Input Layer II. The prepared data-set is called Data-set II and 60 percent of it is used for training process. The architecture in Figure II.4:(A) shows the Input layers I and II consist of tensors of size (Batch Size $\times$ 2 $\times$ 6) and (Batch Size $\times$ 2 $\times$ 3), respectively. The first dimension of all layers are reserved for batch sizes and for the training, the batch size 1 was chosen. In the architecture of the model in Figure II.4:(A), the Input, Hidden and Output Layers I and II together with the number of nodes and type of layers are demonstrated according to Figure II.2:(B).

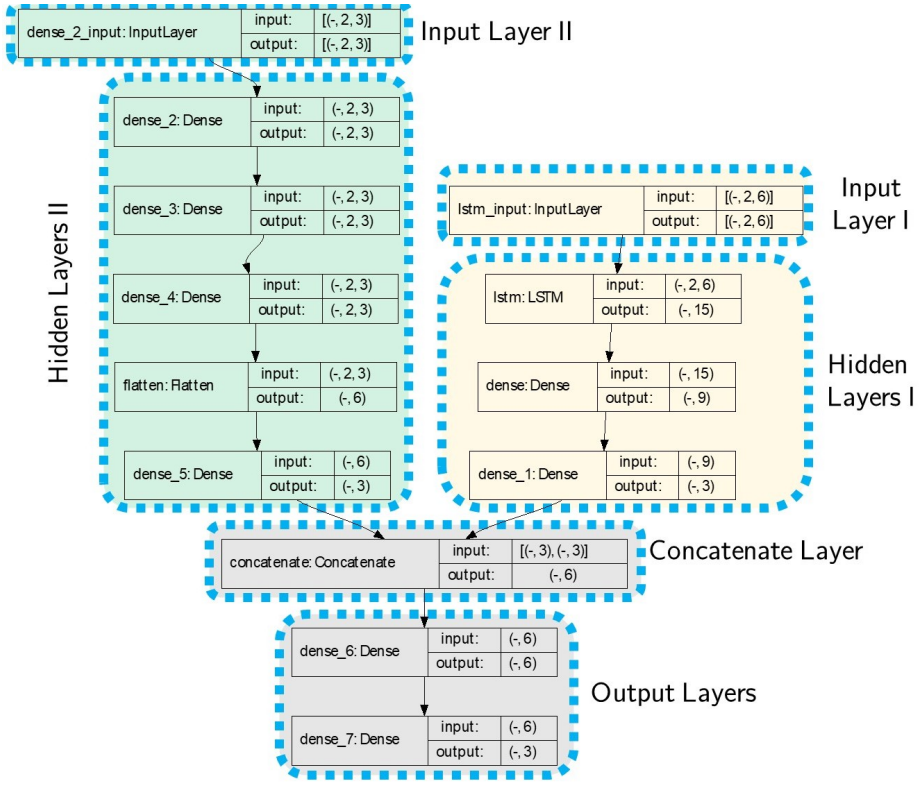
First, unseen data samples in Data-set II are employed to evaluate the model, and tip positions are calculated and the results are shown in Figure II.5:(B). The maximum and mean absolute error are (3.58 mm, 1.70 mm), (1.80 mm, 0.69 mm), and (2.73 mm, 1.41 mm), or in other words, the maximum errors with respect to the length of the manipulator are 0.71%, 0.36%, and 0.54% in $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$-axes, respectively. The RMS errors of the end-effector through Equation (II.2) are shown in Figure II.5:(C).

To evaluate the generalizability of the trained model, different profiles of forces are applied to the model aiming at obtaining different position trajectories for the end-effector as depicted in Figure II.6:(A,D). To fulfill the goal of the second example, the new forces are only used for the evaluation of the trained model by predicting the positions of the end-effector. Results of the prediction are plotted in Figure II.6:(B) and are as follows: The maximum and mean absolute errors are (10.49 mm, 2.61 mm), (5.54 mm, 1.05 mm), and (5.97 mm, 2.83 mm), furthermore, the percentage of the maximum/worst-case errors with respect to the length of the manipulator are 1.90, 1, and 1.08 in the $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$-axes, respectively. The RMS errors of the end-effector through Equation (II.2) are shown in Figure II.6:(C).

In this example, the maximum constant time step for this simulation is 10^{-4} s, to have a convergent numerical solver for the analytical dynamic model. In addition, on average, the time 1.89×10^{-4} s is required for solving the dynamics of each node. In other words, the analytical dynamic model can not achieve any real-time performance for this example. However, the proposed model achieves a real-time performance of the bandwidth of 60.30 Hz on average.

II.4.3 Third Simulation: A Cylindrical Rod with and without External Wrenches

In the last example, we form a semi-circular shape with a cylindrical rod. A force is applied to the middle node — Node 51^{th}— in the $-\boldsymbol{y}-$axis direction for 0.5 s and then the force is removed. Furthermore, the boundary conditions are as follows: the first and last nodes are fixed to the origin and their orientations are



(A)

Radius	1 cm
Number of elements	8
Length	55 cm
Mass density	$10^3 \frac{kg}{m^3}$
Young modulus	10 MPa
Poisson's ratio	0.3
Time step	10^{-4}
Simulation time	16 (S)

(B)

Figure II.4: **(A)**: The model architecture used for the second simulation example. There are two Input layers, the first one is the poses of the node and the second input is the applied forces on the node. The first dimension of inputs and outputs in each layer are unspecified and can vary with the size of batches. **(B)**: Rod properties and simulation parameters used in the second example.

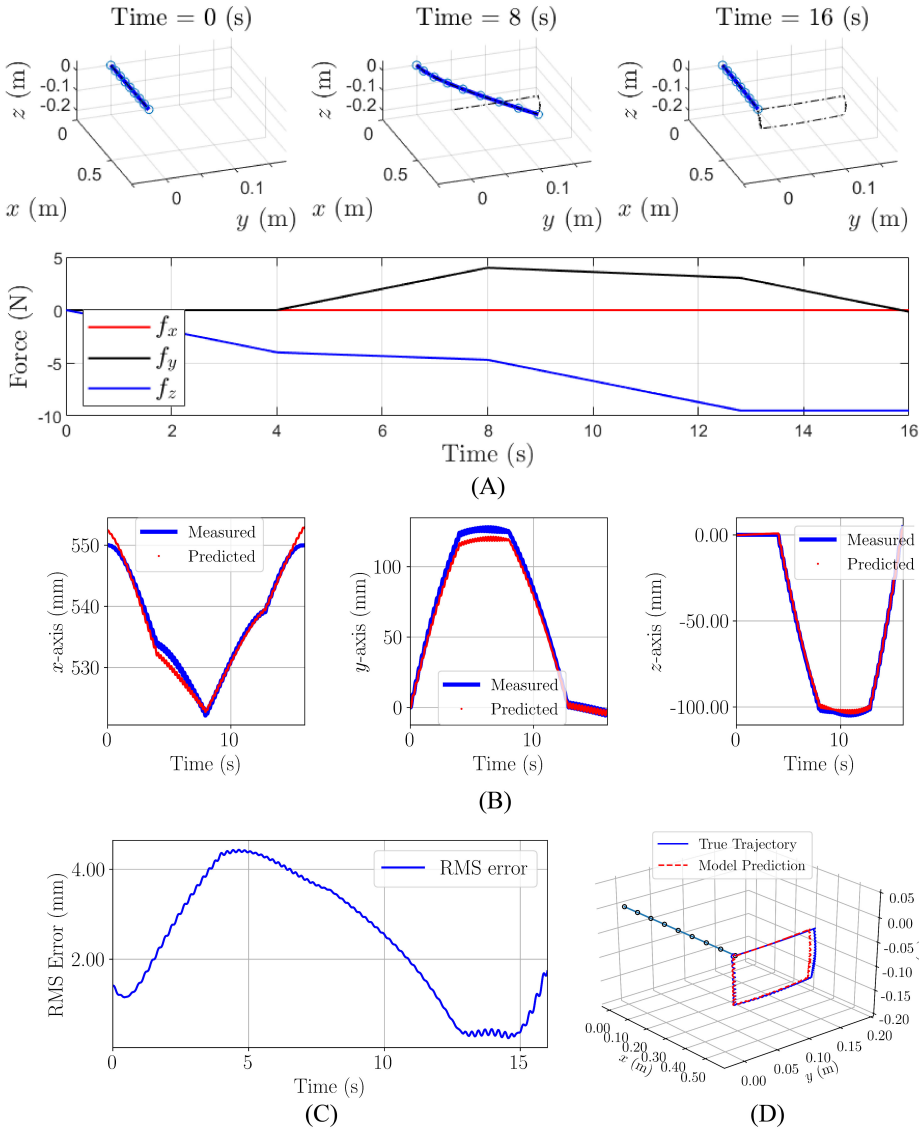


Figure II.5: Second simulation example: **(A)**: Initial, Time-evolved configurations and forces on the last node. **(B)**: Tip positions: calculated from the analytical dynamic model and predicted by the proposed RNN-based model. **(C)** RMS error considering all the axes. **(D)** Predicted and measured trajectories.

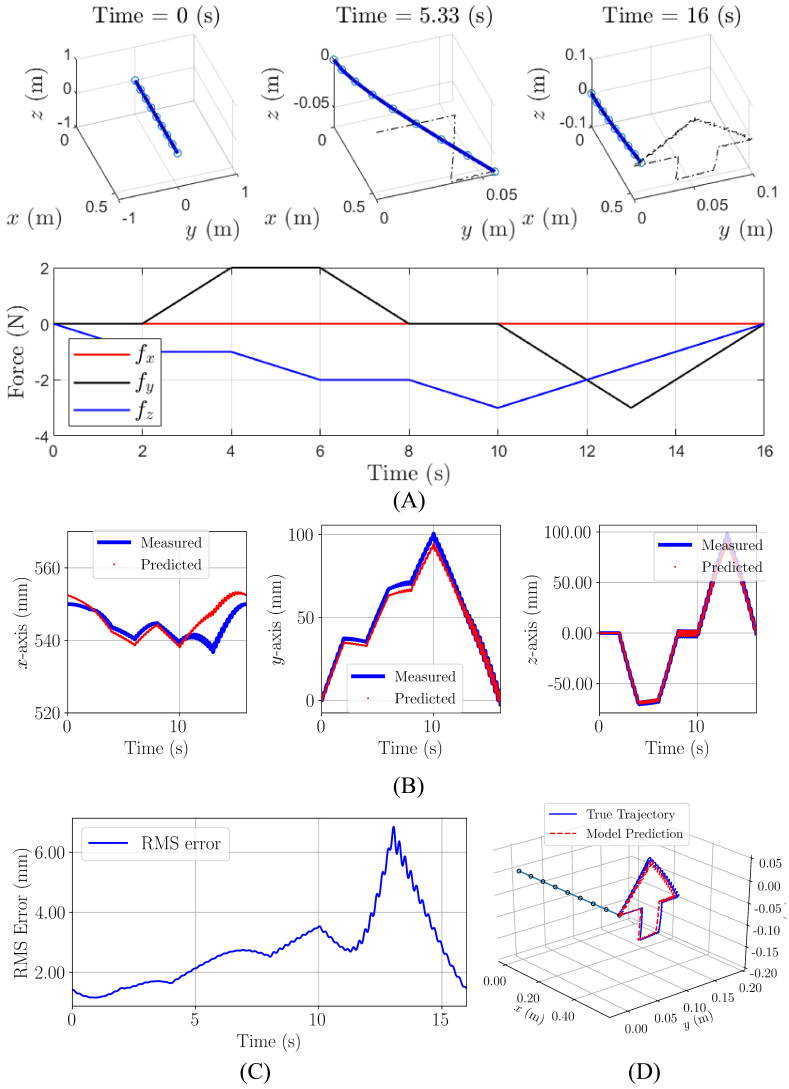


Figure II.6: Evaluation example of Second simulation: **(A)**: Initial, Time-evolved configurations and forces on the end-effector. **(B)**: Tip positions: calculated from the analytical dynamic model and predicted by the proposed RNN-based model. **(C)** RMS error considering all the axes. **(D)** Predicted and measured trajectories.

set to the identity and $\mathtt{R_z}(181.81°)$, respectively, where $\mathtt{R_z}(\theta)$ describes rotation around the z-axis by θ degrees. In this example, the idea is to model the behavior of the rod resulted from applied external forces and internal elastic energy. The structure of the proposed model is given in Figure II.7:(A) and the rod properties and simulation parameters are given in Figure II.7:(B). The initial and a few time-evolved configurations together with the applied forces are given in Figure II.8:(A).

20001 position and force samples are generated from the analytical model for each node. We augment 1-by-3 position vectors of Node 15^{th} and its adjacent nodes at each time step. Therefore, the augmentation results in a 1-by-9 vector. Furthermore, the size of history horizons is chosen to be 2 ($\eta = 1$). Finally, augmented 2-by-9 tensors are obtained for each time steps which are fed to the model as the Input Layer I. The same preparation process are applied for the force data samples which are used as the Input Layer II. The prepared data-set is called Data-set III and 60 percent of the data is used for training process. The architecture in Figure II.7:(A) shows the Input layers I and II consist of tensors of size (Batch Size $\times$ 2 $\times$ 9) and (Batch Size $\times$ 2 $\times$ 3), respectively. The first dimension of all layers are reserved for batch sizes and for the training, the batch size 1 was chosen. In the architecture of the model in Figure II.7:(A), the Input, Hidden and Output Layers I and II together with the number of nodes and type of layers are demonstrated according to Figure II.2:(B).

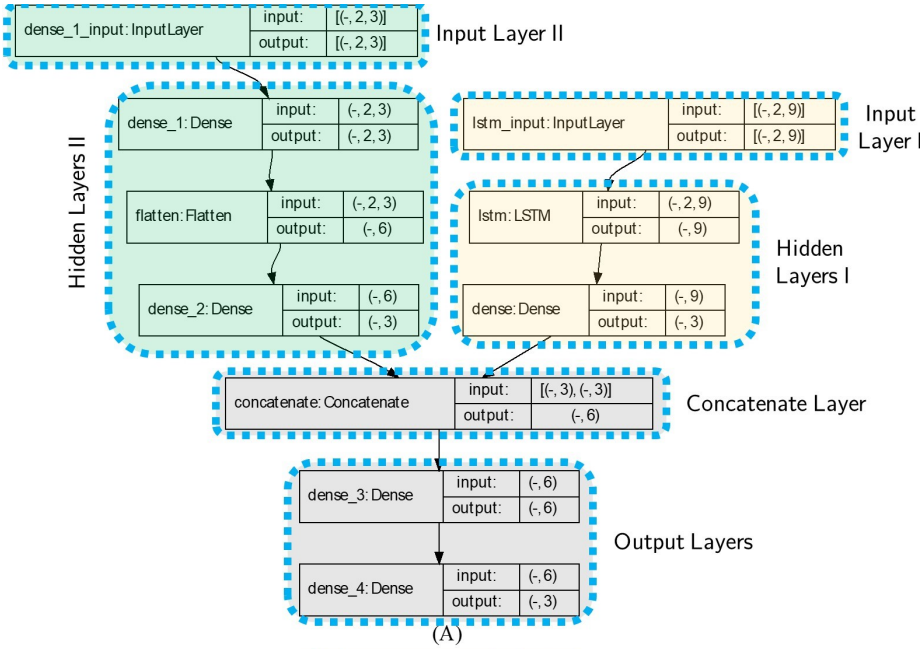
The positions of Node 51^{th} are predicted using seen and unseen data samples in Data-set III and the results are shown in Figure II.8:(B,C). The maximum and mean absolute error are (1.36 mm, 0.78 mm), (0.23 mm, 0.13 mm), and (2.22 mm, 0.81 mm). Furthermore, the maximum/worst-case errors with respect to the length of the manipulator are 0.22%, 0.04%, and 0.37% in the $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$-axes, respectively.

For the evaluation of the trained model and to fulfill the goal of this example, force vector $[0,\ 0,\ -100 \times cos(\pi t)]^T$ mN is applied to Node 51^{th} for $t \in [1, 2]$ s. Results of the prediction are as follows: The maximum and mean absolute errors are (3.73 mm, 1.25 mm), (2 mm, 0.2 mm), and (8.1 mm, 1.2 mm), furthermore, the maximum/worst-case errors with respect to the length of the manipulator are 0.62%, 0.33%, and 1.35% in the $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$-axes, respectively.

In this example, the maximum constant time step for this simulation is 10^{-4} s using the analytical model. In other words, the analytical model does not show a real-time performance since, on average, the time 2.22×10^{-4} s is required for solving the dynamics of each node. On the other hand, the proposed model can achieve a real-time performance of the bandwidth 58.13 Hz on average.

II.5 Experimental Results

This section is devoted to the experimental validation of the presented model. To that end, we fabricated a soft manipulator on which magnetic fields are used to produce necessary forces and torques. Compared to the simulations in which positions are predicted, time-sequence input is composed of orientations of nodes



(A)

Radius	1 cm
Number of elements	100
Length	60 cm
Mass density	$10^3 \frac{kg}{m^3}$
Young modulus	10 KPa
Poisson's ratio	0.3
Time step	10^{-4}
Simulation time	2 (s)

(B)

Figure II.7: **(A)**: The model architecture used for the third simulation example. There are two Input layers, the first one is the poses of the node and the second input is the applied forces on the node. The first dimension of inputs and outputs in each layer are unspecified, and can vary with the size of batches. **(B)**: Rod properties and simulation parameters used in the third example.

in the experiment. Furthermore, to show the performance of the algorithm, results from the presented method and a Cosserat rod-based theoretical model are compared to show the efficiency of the proposed RNN-based model. The Cosserat rod model of the soft manipulator is detailed in the appendix.

II.5.1 Soft Continuum Manipulator

A soft continuum manipulator is fabricated from a urethane rubber Polymer Matrix Composite 770 (PMC-770, Smooth-On Inc., USA) and neodymium (NdFeB) block magnets whose dimensions are given in Figure II.9:(A). When the manipulator is subjected to an external magnetic field, the embedded magnets

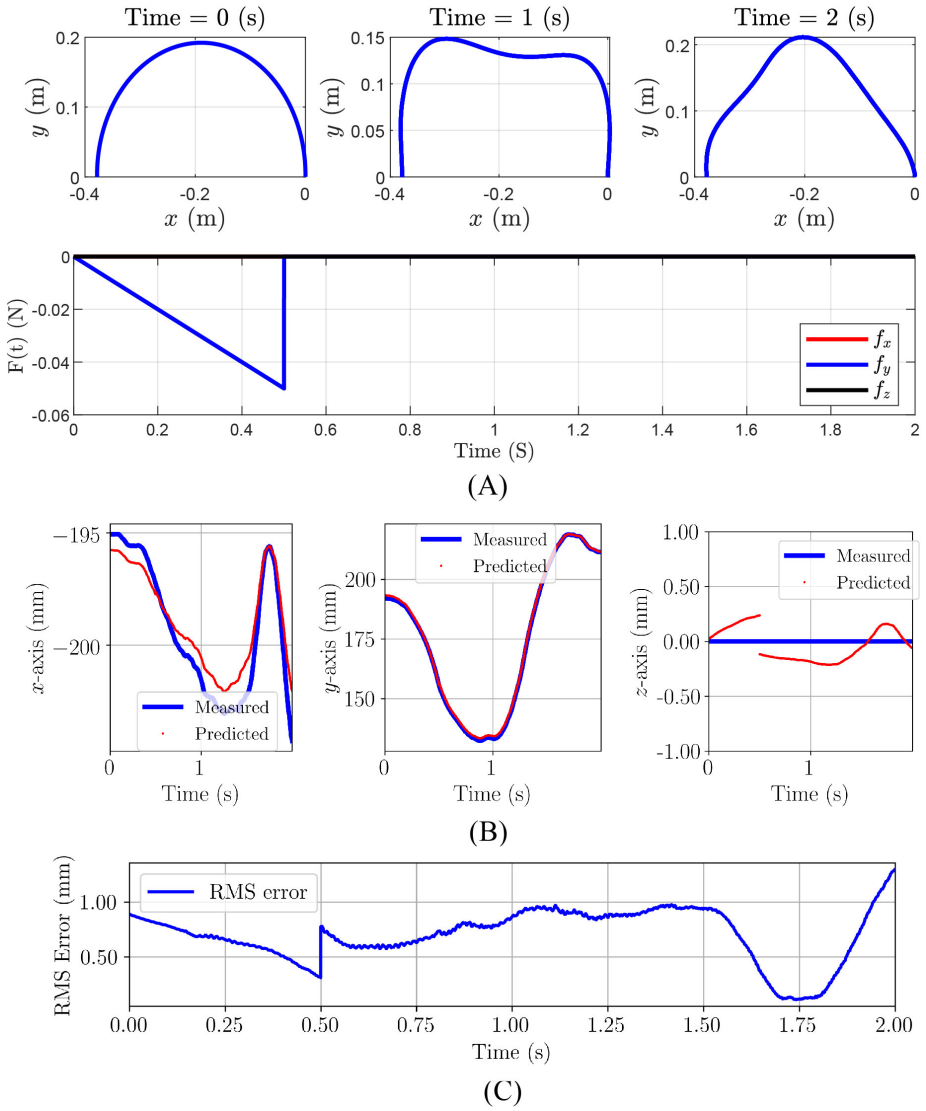


Figure II.8: Third simulation example: **(A)**: Initial, Time-evolved configurations and forces applied on the rod. Positions of the middle node, where the force is applied, are measured and predicted. **(B)**: Measured position calculated from analytical dynamic model and predicted by the proposed RNN-based model in the $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$-axes are given. **(C)** RMS error considering all the axes.

experience forces and torques. This causes the flexible portions of the manipulator comprised of the PMC to undergo elastic deformation.

The PMC-770 has a density $\rho = 1000$ kg/m^3, Young modulus $E = 2.5$ MPa,

and Poisson ratio $\nu = 0.5$. The distal and proximal NdFeb magnets have grades N45 and N42, respectively. In addition, they have density $\rho = 7000$ kg/m^3, Young modulus $E = 41.4$ GPa, and Poisson ratio $\nu = 0.3$. It should be pointed out that Young's modulus and densities of the soft manipulator constituent materials were determined using a combination of supplier data and experiments until theoretical results (predicted by the Cosserat rod model) would resemble the experiment results. The magnitude of the magnetic dipoles carried by the manipulator was calculated from the magnets volume and manufacturer-supplied residual flux density.

II.5.2 Experimental Setup

The experimental setup consists of 6 stationary electromagnets surrounding a spherical workspace of 100 mm diameter [Sik+17]. Figure II.9:(B) shows the setup of the experiment. In addition, the final shape of manipulator has been segmented and is shown in the workspace. The continuum manipulator is suspended horizontally (along $\boldsymbol{x}$) in the workspace and actuated to move in a plane, steering the magnets by manipulating the magnetic field generated by the electromagnets. Orientations are represented using the axis-angle notation. Let $\boldsymbol{k}_m \in \mathbb{R}^3$ and $\phi_m \in \mathbb{R}$ denote the axis- and angle-of-rotation, respectively, where $m = 1, 2$ denotes the magnet index counting from the manipulator base. In the 2D experiment, $\boldsymbol{k}_m = \boldsymbol{y}$, and ϕ_m is defined relative to $\boldsymbol{z}$.

Figure II.9:(C) represents the shape reconstruction of the soft manipulator through images coming from two Dalsa Genie Nano C1940 Red-Green-Blue (RGB) cameras (TeledyneDalsa, Waterloo, ON, Canada). The flexible PMC-770 and rigid NdFeb magnets were colored blue and red, respectively. The RGB cameras (horizontal and vertical) that formed a stereo vision setup recorded the workspace during experiments. First, we discretize the actuation workspace into voxels. The silhouette of the continuum manipulator is segmented as binary masks and the manipulator body represented as a 3D spatial point cloud. The manipulator centerline is approximated by $N \in \mathbb{N}$ discrete segments. A simple iterative shape reconstruction algorithm [Sik+19] moves through the voxels to represent the the manipulator centerline with N discrete points ($\{\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_N\}$) as a function of centerline parameter $s \in [0, L]$. To be specific, with knowledge of the RGB-camera frames, the points are projected onto each camera image. If a point is projected onto both binary masks, the point falls within the manipulator. This process is repeated for all voxels. Subsequently a 3D polynomial fit ($\boldsymbol{P}(s)$) is made through the points. We assume that magnetically exerted forces and torques are insufficient for the manipulator extension along the centerline, and therefore assume constant positions of the magnets along the centerline, $s_m \in (0, L)$. The measured magnet position is thereafter obtained from the polynomial fit ($\boldsymbol{p}_m = \boldsymbol{P}(s_m)$), and its orientation from the local gradient of the polynomial fit ($\partial_s \boldsymbol{P}(s_m)$) relative to a reference $\boldsymbol{z}$ axis,

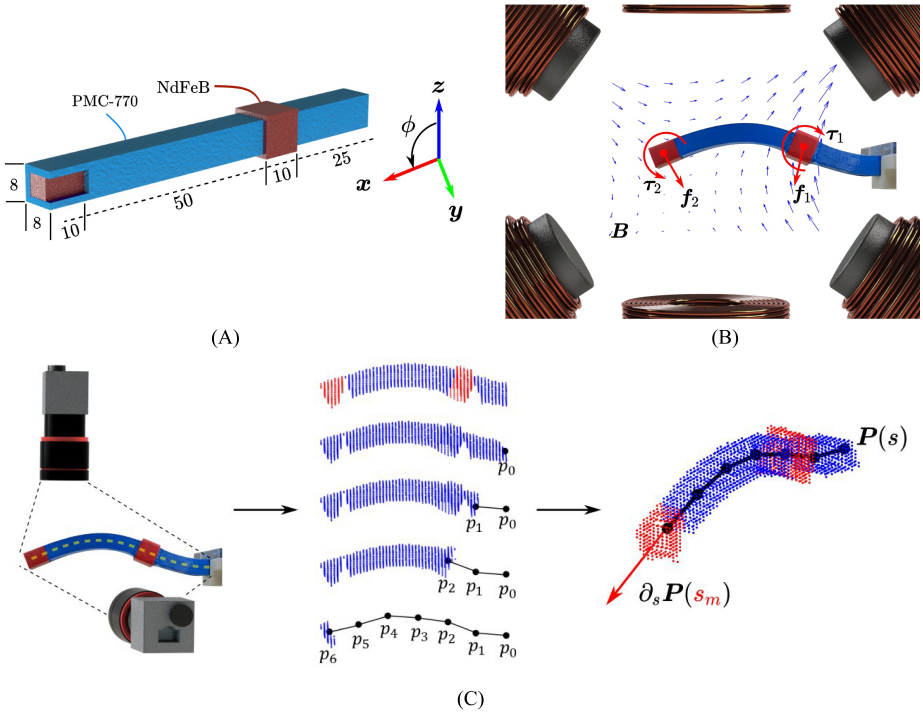


Figure II.9: **(A)**: Polymer matrix composite 770 (PMC-770) beam continuum manipulator with embedded neodymium (NdFeb) magnets located at tip and intermediate positions. Dimensions are given in millimeter. **(B)**: Experimental setup consists of 6 stationary electromagnets and contains a segmented photograph of the final shape manipulator. The flexible PMC-770 and rigid NdFeb sections of the manipulator are blue and red, respectively. Six electromagnets generate a magnetic field ($\boldsymbol{B}$) in the workspace, exerting torques and forces ($\boldsymbol{\tau}_m, \boldsymbol{f}_m$, $m = 1, 2$) on the magnets, which deforms the continuum manipulator to its final shape at the time $t = 340$ s. **C**: Representation of the shape reconstruction algorithm used for shape feedback. The manipulator is recorded with a stereo vision setup. The manipulator body is represented by a 3D spatial point cloud. The manipulator centerline, characterized by parameter $s \in [0, L]$, is approximated by $N + 1$ points ($\{\boldsymbol{p}_0, \ldots, \boldsymbol{p}_N\}$). A 3D polynomial fit ($\boldsymbol{P}(s)$) is made through the points, and the magnet orientation at an assumed constant centerline position s_m derived from the local gradient of the polynomial fit.

$$\phi_m = \cos^{-1}\left(\boldsymbol{z} \cdot [\partial_s \boldsymbol{P}(s_m)]^\wedge\right),$$
$$\boldsymbol{k}_m = [\boldsymbol{z} \times \partial_s \boldsymbol{P}(s_m)]^\wedge$$

where $[\cdot]^\wedge$ represents a normalization. Furthermore, in the experiments performed for this study, camera occlusions did not occur.

The magnetic torques and forces were computed from the magnets position $\boldsymbol{p}_m$, magnets dipole moments $\boldsymbol{\mu}_m \in \mathbb{R}^3$, and electromagnet currents $\boldsymbol{I_C} \in \mathbb{R}^6$. The magnets position and orientations were obtained from the stereo vision setup. Afterwards, the orientations are used to compute magnets dipoles. To compute the magnetic field, each electromagnet is associated with a unit-current field and field gradient map ($\boldsymbol{\beta}_i(\boldsymbol{p}) \in \mathbb{R}^3$ and $\boldsymbol{\beta}_{\nabla,i}(\boldsymbol{p}) \in \mathbb{R}^{3\times3}$, $i = 1, \ldots, 6$), which computes the unit-current contribution of the electromagnet to the field at field point $\boldsymbol{p}$. We define a map $\mathbb{G}(\boldsymbol{\beta}_{\nabla,i}) : \mathbb{R}^{3\times3} \to \mathbb{R}^5$ which takes the five independent gradient terms of the field [PN15]. The field (gradient) at magnet position $\boldsymbol{p}_m$ is then given by the superposition principle

$$\begin{aligned}\boldsymbol{B}_m &= \begin{bmatrix}\boldsymbol{\beta}_1(\boldsymbol{p}_m), & \ldots, & \boldsymbol{\beta}_6(\boldsymbol{p}_m)\end{bmatrix} \boldsymbol{I_C} \\ \mathbb{G}(\boldsymbol{B}_{\nabla,m}) &= \begin{bmatrix}\mathbb{G}(\boldsymbol{\beta}_{\nabla,1}(\boldsymbol{p}_m)), & \ldots, & \mathbb{G}(\boldsymbol{\beta}_{\nabla,6}(\boldsymbol{p}_m))\end{bmatrix} \boldsymbol{I_C}.\end{aligned}$$

The torques and forces exerted on the magnets due to the field is given by

$$\begin{aligned}\boldsymbol{\tau}_m &= \boldsymbol{\mu}_m \times \boldsymbol{B}_m, \\ \boldsymbol{f}_m &= \nabla(\boldsymbol{\mu}_m \cdot \boldsymbol{B}_m)\end{aligned} \quad \Rightarrow \quad \begin{aligned}\boldsymbol{\tau}_m &= [\boldsymbol{\mu}_m]_\times \boldsymbol{B}_m \\ \boldsymbol{f}_m &= \mathbb{M}(\boldsymbol{\mu}_m)\mathbb{G}(\boldsymbol{B}_{\nabla,m})\end{aligned}$$

where $\mathbb{M}(\boldsymbol{\mu}_m) : \mathbb{R}^3 \times \mathbb{R}^{3\times5}$ represents a map of the field independent spatial gradients to forces on the dipole $\boldsymbol{\mu}_m$ [PN15]. The applied magnetic forces and torques together with the initial and a few time-evolved configurations are shown in Figure II.10.

For modeling, we consider three nodes located at the locations of the proximal and distal magnets, and the clamped end of the rod. It should be pointed out that the performance of the proposed RNN-based model, unlike conventional algorithms, is independent of the number of nodes considered for the whole manipulator. Therefore, it is sufficient to model points of interest. The idea is to independently manipulate each magnet (actuation point). However, the setup provides us with 8 degrees of freedom, meaning that positions and orientations (12 degrees of freedom) cannot be manipulated at the same time. Therefore, we carried out the experiment to achieve only orientation control.

669 1-by-3 position samples and 1-by-6 augmented wrench samples (i.e., $[\tau, f]^T$) for the both magnets are obtained. By choosing the size of history horizon 2 ($\eta = 1$), the augmented 2-by-6 position tensors are reshaped for each time step and fed to the model as the Input Layer I. The same preparation process is applied for the force data samples which are used as the Input Layer II. The prepared data-set is called Data-set IV and 60 percent of the data is

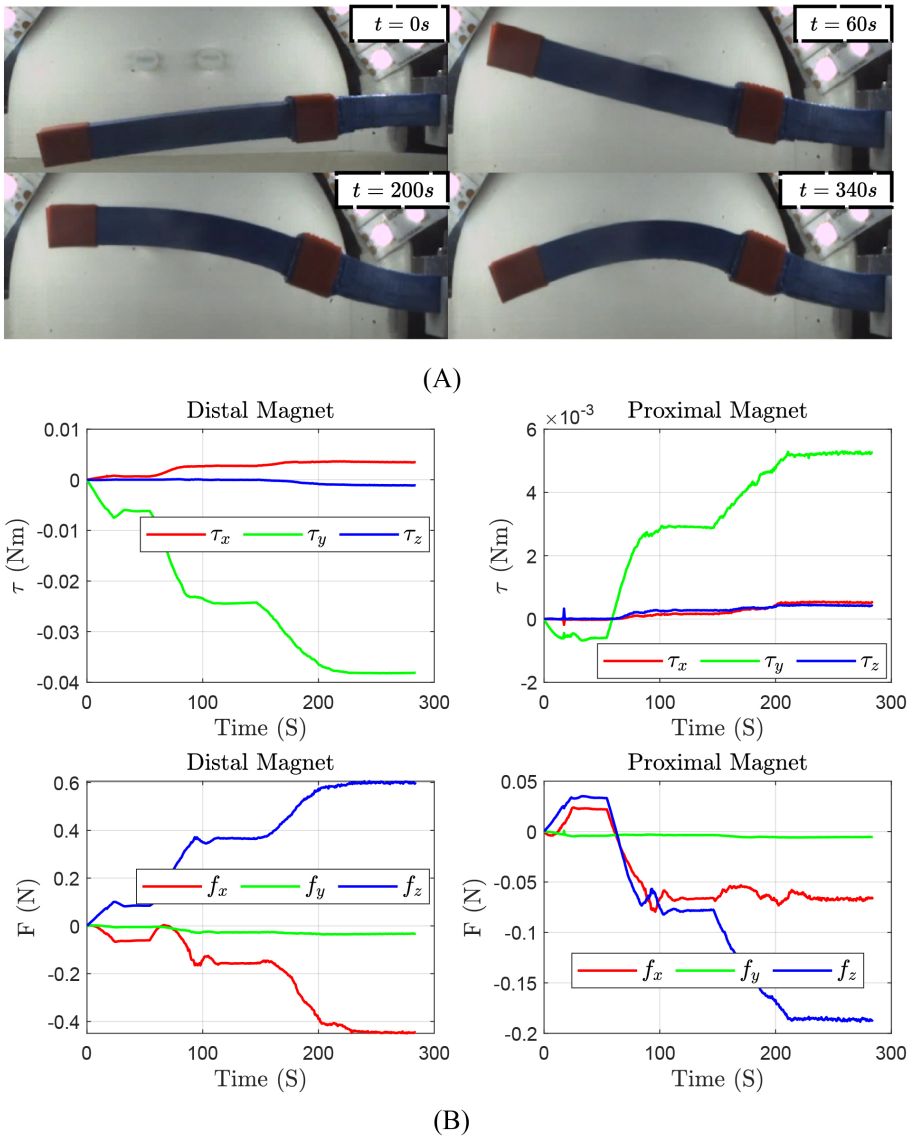


Figure II.10: **(A)**: Initial and time-evolved configurations. **(B)**: Applied torques and forces on the distal and proximal magnets.

used for training process. We suggest the same model for both moving nodes and the architecture of the model is depicted in Figure II.11 which is the same for the proximal and distal nodes. The architecture shows the Input layers I and II consist of tensors of size (Batch Size $\times$ 2 $\times$ 6). The first dimension of all layers are reserved for batch sizes and for the training, the batch size 1 was chosen. In the architecture of the model in Figure II.11, the Input, Hidden and Output Layers I and II together with the number of nodes and type of layers are demonstrated according to Figure II.2:(B).

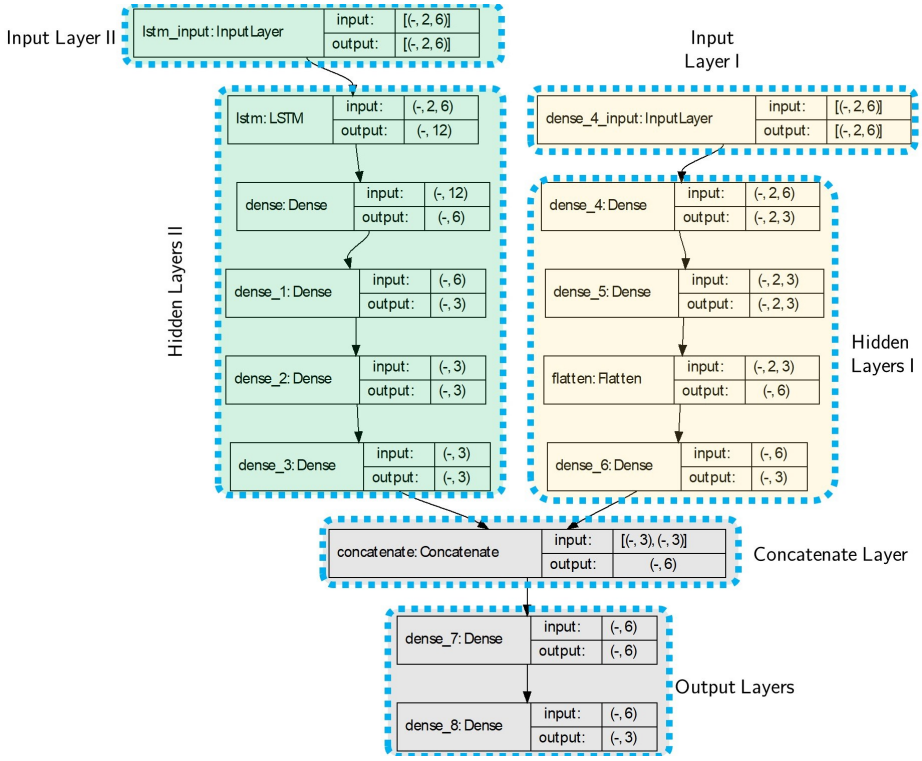


Figure II.11: The model architecture used for the experiment. There are two Input layers, the first one is the poses of the node and the second input is the applied forces on the proximal node. The first dimension of inputs and outputs in each layer are unspecified, and can vary with the size of batches.

II.5.3 Results

The distal and proximal node rotations are predicted both by Cosserat rod model and the proposed model, and the results are shown in Figure II.12. Also, the maximum and mean absolute errors are stated in an ordered pair in Table II.1.

The computation time required to find a solution of the manipulator statics from a Boundary Value Problem (BVP) with Cosserat rod theory depends

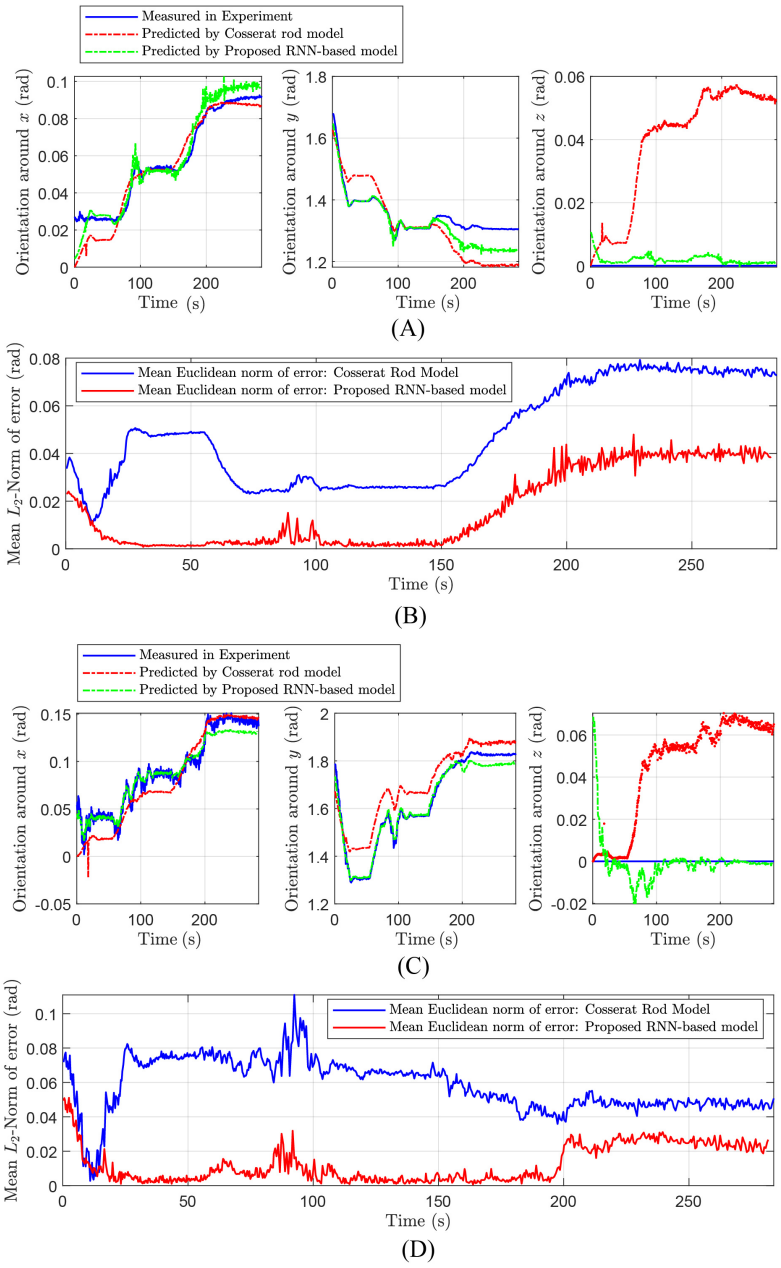


Figure II.12: **(A)**: Measured and predicted orientations of tip/distal magnet by Cosserat rod model and proposed RNN-based model. **(B)**: Mean Euclidean norm of error for distal/tip magnet resulted from Cosserat rod model and proposed RNN-based model. **(C)**: Measured and predicted orientations of middle/proximal magnet by Cosserat rod model and proposed RNN-based model. **(D)**: Mean Euclidean norm of error for middle/proximal magnet resulted from Cosserat rod model and proposed RNN-based model.

Results for distal node	$\boldsymbol{x}$**-axis**	$\boldsymbol{y}$**-axis**	$\boldsymbol{z}$**-axis**
Proposed RNN-based model	$(1.27^{\circ}, 0.23^{\circ})$	$(4.69^{\circ}, 1.27^{\circ})$	$(0.60^{\circ}, 0.10^{\circ})$
Cosserat rod model	$(3.56^{\circ}, 0.82^{\circ})$	$(10.55^{\circ}, 5.03^{\circ})$	$(4.02^{\circ}, 2.22^{\circ})$
Results for proximal node	$\boldsymbol{x}$**-axis**	$\boldsymbol{y}$**-axis**	$\boldsymbol{z}$**-axis**
Proposed RNN-based model	$(1.13^{\circ}, 0.31^{\circ})$	$(3.30^{\circ}, 0.89^{\circ})$	$(3.93^{\circ}, 0.31^{\circ})$
Cosserat rod model	$(1.55^{\circ}, 0.36^{\circ})$	$(7.22^{\circ}, 3.51^{\circ})$	$(3.29^{\circ}, 1.99^{\circ})$

Table II.1: The maximum and mean absolute errors around the $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$-axes in ordered pairs for the distal and proximal nodes.

on the quality of the initial solution guess, i.e, $\boldsymbol{n}(s)$ and $\boldsymbol{m}(s)$ at $s = 0$, the tolerable error ($E \in \mathbb{R}$), and the number of nodes ($N \in \mathbb{N}$) used to discretize the manipulator.

A tolerable error describes the error between the distal internal forces and moments obtained from forward integration which are called $\boldsymbol{n}_f^d$ and $\boldsymbol{m}_f^d$, and distal boundary condition, i.e., $\boldsymbol{n}_b^d$ and $\boldsymbol{m}_b^d$. The tolerable error can be written as $\left\| \left[\boldsymbol{n}_f^d - \boldsymbol{n}_b^d, \quad \boldsymbol{m}_f^d - \boldsymbol{m}_b^d\right] \right\|_2 \leq E$.

Decreasing the tolerable error increases the solution accuracy, but potentially requires more time to solve convex optimizations for the BVP. Increasing the number of nodes is necessary to describe complex manipulator geometries, but should be chosen to minimize the required steps during forward integration.

To visualize how the required computation time changes with the number of nodes and the tolerable error, multiple simulations were performed by assigning known torques $\boldsymbol{\tau}_m$ and forces $\boldsymbol{f}_m$ for $m = 1, 2$, to the manipulator, and finding a valid solution from solving the BVP. Changes to tolerable errors (E) and number of nodes (N) were made manually. For example, an error of 2% in the initial solution guess was obtained by multiplying the valid solution with 0.98. After each change the BVP was solved again fifty times. The obtained mean and standard deviation of the computation times are shown in Figure II.13. By taking into account all the aforementioned variables, i.e., number of nodes and the tolerable error, the Cosserat rod model is capable of achieving real-time performances of the bandwidths between 8.33 Hz and 50 Hz.

Figure II.13:(A) shows how the computation time required for solving a solution to the BVP changes with decreasing tolerable error (E) and increasing percentage errors from a valid solution (at 0%), for a constant number of nodes. Also, Figure II.13:(B) shows how the computation time required for solving a solution to the BVP changes with an increasing number of nodes (N) and increasing percentage errors from a valid solution, for a constant tolerable error. However, It should be mentioned that the proposed RNN-based model shows a real-time performance with a bandwidth of 60.75 Hz on average for the given architecture in Figure II.7, number of epochs = 25, and batch size = 1. In addition, Figure II.14:(A) demonstrates the computation bandwidth required for the prediction of the next step using the trained model with a different number of LSTM units and a different size of time history horizons in Data-set IV. The

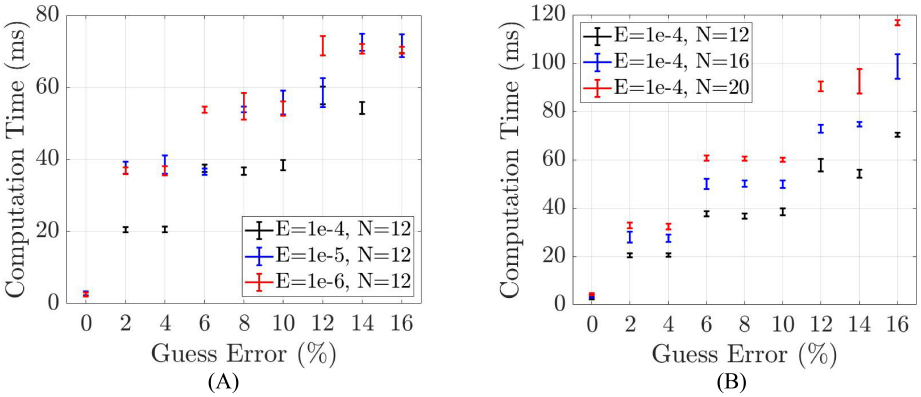


Figure II.13: **(A)**: Computation time required for solving a solution to the BVP changes with decreasing tolerable error (E) for a constant number of nodes. **(B)**: Computation time required for solving a solution to the BVP changes with an increasing number of nodes (N) and increasing percentage errors from a valid solution, for a constant tolerable error.

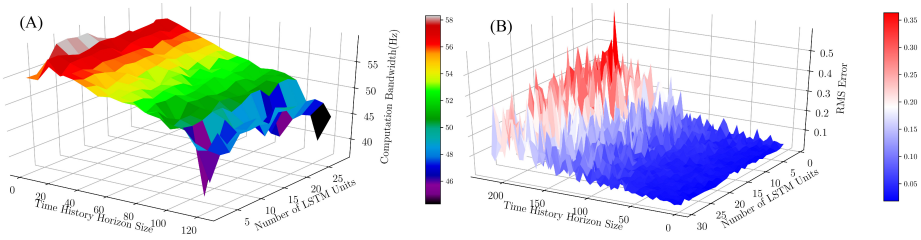


Figure II.14: **(A)**: Computation bandwidth (Hz) obtained with respect to different number of LSTM units and size of time history horizon (for number of epochs = 25 and batch size = 1). **(B)**: RMS Error (mm) of the prediction on Data-set IV using trained models with different number of LSTM units and size of time history horizons (for number of epochs = 25 and batch size = 1).

figure suggests that computation bandwidths are fairly unchanged with the number of LSTM units; however, increasing the length of time history reduces bandwidth. The optimal region maximizing the bandwidth is approximately with time history size in $(2, 20)$ and LSTM unit size in $(5, 15)$. Figure II.14:(B) suggests that RMS error of the prediction decreases by increasing the number of LSTM units and the optimal area minimizing RMS errors is approximately with time history size in $(2, 20)$ and LSTM unit size in $(20, 25)$.

To sum up, this experiment demonstrates that not only can the presented RNN-based model outperform classical modeling approaches such as the Cosserat rod model, but also it shows possibilities to use the model in practice for closed-loop control applications.

II.6 Discussion

This work suggests a distributed architecture for modeling complex dynamical systems by using multiple light-weight RNN-based models. As a result, the architecture would be easier to design and debug, and also benefits from faster convergence compared to one large network. Furthermore, large networks may take longer times to be trained, and they may not show an acceptable performance and readjusting (hyper-)parameters and restarting the training process might be necessary.

Increasing the size of history horizons in training stages may reduce the error to some extent, but on the other hand, it makes the model slower. Based on conventional dynamical models, the length of the history size should be at least 2. To reach a state-of-the-art performance, i.e., having less error and faster model simultaneously, one may prefer varied batch sizes in the training and run-time phases. As a suggestion, we can use different batch sizes for training and run-time stages. A model can be trained with appropriate batch sizes such that the model performance suits the given criteria. Afterward, one can create a new network with the pre-trained weights compiled with a batch size of 1.

The performance, i.e., the convergence and stability, of the presented algorithm in this paper, unlike conventional algorithms, is independent of the number of nodes considered for the whole manipulator. To be specific, in the analytical model, there might be a need for several discretization nodes to achieve a convergent solution with a specific tolerable error; however, in the RNN-based model, only specific points/ points of interests (e.g., two actuation points in the experiment) are considered. In other words, in the experiment, 13 nodes (4 for each flexible subsection and 2 for each magnet, and 1 for the base) were chosen for solving the Cosserat rod model, but two nodes were selected for the RNN-based model. However, the complexity of dynamical systems (i.e., PDEs) affects the complexity of the architecture used in the RNN-based model, i.e., the number of layers and LSTM units and generally how deep the model is. Nevertheless, the suggested model suits parallel implementation and can benefit from a high bandwidth for closed-loop control applications. Furthermore, the architectures of the proposed RNN-based model can be optimized by reducing the number of layers and trainable parameters to maximize the achievable bandwidths.

The evaluations showed that incorporating poses of adjacent nodes and also wrenches as a separated input might help to have, to some extent, a generalizable model rather than just purely learning the structure of data. However, supervised learning methods likely tend to preserve structure of data, and these models might not entirely respect underlying physics (conservation laws). In other words, these methods might not be wholly physics-aware and applicable for untrained/unprecedented dynamics or geometries without any adjustment, re-training, or using techniques such as transfer learning, etc. One possible and interesting solution [LLF98; PU92; RPK19] to overcome this problem and move toward fully physics-aware neural networks is revisiting lost functions for the training process. To be specific, it is mentioned in Problem Statement Section that the idea is finding solutions for PDEs given in Equation (II.1), i.e., $\Lambda(x, t)$

and $\phi(x,t)$ for sufficiently large number (e.g., N_f) of pair $(x_i,t_i) \in (0,L)\times[0,T]$ in which L is the unreformed length of the manipulator and parameter T is a user-defined time. Considering Equation (II.1), a neural network can be learned by minimizing the mean squared error loss

$$\frac{1}{N_f}\left(\sum_{i=0}^{N_f}\left\|\boldsymbol{J}\boldsymbol{\omega}_{t_i}+\boldsymbol{\omega}\times\boldsymbol{J}\boldsymbol{\omega}+\boldsymbol{n}\times\boldsymbol{\Lambda}^{-1}\boldsymbol{\phi}_{x_i}-\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_{x_i}\times\boldsymbol{m}-\boldsymbol{m}_{x_i}-\boldsymbol{\Lambda}^{-1}\boldsymbol{\tau}\right\|^2_{\{x_i,t_i\}}\right.$$
$$\left.+\sum_{i=0}^{N_f}\left\|M\boldsymbol{\phi}_{t_it_i}-\boldsymbol{\Lambda}(\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_{x_i}\times\boldsymbol{n})-\boldsymbol{\Lambda}\boldsymbol{n}_{x_i}+\boldsymbol{f}^{nc}-\boldsymbol{f}\right\|^2_{\{x_i,t_i\}}\right)$$

This modified loss function enforces the structure imposed by Equation (II.1) for large number (e.g., N_f) of pair $(x_i,t_i)\in(0,L)\times[0,T]$ and the trained neural network will be aware of governing PDEs.

II.7 Conclusion

This paper describes an approach for the real-time prediction of dynamics for general continuum soft manipulators, based on machine learning techniques and Lie group variational integration methods. Poses of a soft, polymer-based manipulator, in the presence of conservative and non-conservative wrenches, are predicted and validated experimentally. The comparison results of the proposed model and a well-known model for continuum manipulators, i.e., Cosserat rod theory, are also provided, revealing the practical effectiveness of the proposed model. The presented method can be extended to different soft robots with different shapes and materials. In addition, training of physics-aware neural networks for solving PDEs and the procedure of a model-based controller design are topics of research to be studied as future work.

Appendix II.A Cosserat Rod Theory

The Cosserat rod model of the manipulator assigns a position $\boldsymbol{p}(s)\in\mathbb{R}^3$, orientation quaternion $\boldsymbol{q}(s)=(q_r,\boldsymbol{q}_i)\in\mathbb{R}^4$, internal force $\boldsymbol{n}(s)\in\mathbb{R}^3$, and internal moment $\boldsymbol{m}(s)\in\mathbb{R}^3$ to a material cross-section at centerline position $s\in[0,L]$, where $L\in\mathbb{R}^+$ is the length of the manipulator, giving a material state vector $\boldsymbol{y}=[\boldsymbol{p}^T,\boldsymbol{q}^T,\boldsymbol{n}^T,\boldsymbol{m}^T]^T$. A set of thirteen ordinary differential equations

describe how the state vector evolves between centerline positions [EPN17]

$$\begin{aligned}
\boldsymbol{p}' &= \boldsymbol{R}(\boldsymbol{q})\boldsymbol{v}, \\
\boldsymbol{q}' &= \frac{1}{2}\begin{bmatrix} -\boldsymbol{q}_i^T \\ q_r\boldsymbol{I}_3 - [\boldsymbol{q}_i]_\times \end{bmatrix}\boldsymbol{R}(\boldsymbol{q})\boldsymbol{u}, \\
\boldsymbol{n}' &= -\boldsymbol{f}, \\
\boldsymbol{m}' &= -\boldsymbol{p}' \times \boldsymbol{n} - \boldsymbol{\tau}, \\
\boldsymbol{v} &= K_s^{-1}\boldsymbol{R}(\boldsymbol{q})^T\boldsymbol{n} + \hat{\boldsymbol{v}}, \\
\boldsymbol{u} &= K_b^{-1}\boldsymbol{R}(\boldsymbol{q})^T\boldsymbol{m} + \hat{\boldsymbol{u}},
\end{aligned}$$

where $\boldsymbol{p}' \equiv \partial_s \boldsymbol{p}$, $[\cdot]_\times : \mathbb{R}^3 \to \mathbb{R}^{3\times3}$ a mapping to a skew-symmetric matrix, $\boldsymbol{R}(\boldsymbol{q}) \in SO(3)$ the rotation matrix associated with orientation quaternion $\boldsymbol{q}$, $K_s, K_b \in \mathbb{R}^{3\times3}$ diagonal shear and bending stiffness matrices, $\boldsymbol{I}_3 \in \mathbb{R}^{3\times3}$ a unit matrix, $\boldsymbol{v}(s) \in \mathbb{R}^3$ and $\boldsymbol{u}(s) \in \mathbb{R}^3$ the material strain and bending, and $\hat{\boldsymbol{v}}(s) = [0, 0, 1]^T$ and $\hat{\boldsymbol{u}}(s) = [0, 0, 0]^T$ the intrinsic material strain and curvature. External forces $\boldsymbol{f}(s) \in \mathbb{R}^3$ and torques $\boldsymbol{\tau}(s) \in \mathbb{R}^3$ determine the shape of the manipulator [Ant95; RW11]. The manipulator is subject to a distributed gravity force (not shown in Fig. II.9) and magnetically exerted distributed forces $\boldsymbol{f}_m$ and torques $\boldsymbol{\tau}_m$ due to interaction of the magnets with the magnetic field $\boldsymbol{B}_m$, where $m = 1, 2$ denotes the magnet index from the base of manipulator. Given the exerted torques and forces, the shape of the manipulator is solved as a Boundary Value Problem (BVP). The base of the manipulator is fixed to a rigid base with constant position $\boldsymbol{p}_0$ and orientation $\boldsymbol{q}_0$, and its distal tip is free with constant internal force $\boldsymbol{n}_L^d$ and moment $\boldsymbol{m}_L^d$. The proximal and distal boundary conditions are then formulated as follows, assuming no tip wrench, we have $\mathbb{P}(\boldsymbol{y}_0) = [\boldsymbol{p}_0^T, \boldsymbol{q}_0^T]^T$ and $\mathbb{D}(\boldsymbol{y}_L) = [\boldsymbol{n}_L^{d\,T}, \boldsymbol{m}_L^{d\,T}]^T = \boldsymbol{0}$. The BVP is solved with a forward integration using an explicit Runge-Kutta fourth order method, and convex optimization using Levenberg-Marquardt [TAR19]. The unknown proximal state parameters $\boldsymbol{\xi} = [\boldsymbol{n}_0^T, \boldsymbol{m}_0]^T$ are guessed and subject to the optimization where $N \in \mathbb{N}$ is the number of discrete steps along s. Then $\boldsymbol{y}_N$ are the manipulator distal state parameters obtained from the forward integration using an explicit Runge-Kutta fourth order method. The error between desired distal boundary condition $\mathbb{D}(\boldsymbol{y}_L)$ and $\mathbb{D}(\boldsymbol{y}_N)$ determines if the solution ($\boldsymbol{\xi}$) is accepted.

References

[Ans+17] Ansari, Y. et al. “Towards the development of a soft manipulator as an assistive robot for personal care of elderly people”. In: *International Journal of Advanced Robotic Systems* vol. 14, no. 2 (Apr. 2017), p. 172988141668713.

[Ant95] Antman, S. S. *Nonlinear problems of elasticity*. Applied mathematical sciences v. 107. New York: Springer-Verlag, 1995.

[Bur+13] Burgner, J. et al. “Debulking from within: a robotic steerable cannula for intracerebral hemorrhage evacuation”. In: *IEEE Transactions on Biomedical Engineering* vol. 60, no. 9 (Sept. 2013), pp. 2567–2575.

[Dem+15] Demoures, F. et al. “Discrete variational Lie group formulation of geometrically exact beam dynamics”. en. In: *Numerische Mathematik* vol. 130, no. 1 (May 2015), pp. 73–123.

[EHJ17] E, W., Han, J., and Jentzen, A. “Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations”. In: *arXiv:1706.04702 [cs, math, stat]* (June 2017). arXiv: 1706.04702.

[EPN17] Edelmann, J., Petruska, A. J., and Nelson, B. J. “Magnetic control of continuum devices”. In: *The International Journal of Robotics Research* vol. 36, no. 1 (Jan. 2017), pp. 68–85.

[GDS19] Grazioso, S., Di Gironimo, G., and Siciliano, B. “A geometrically exact model for soft continuum robots: the finite element deformation space formulation”. In: *Soft Robotics* vol. 6, no. 6 (Dec. 2019), pp. 790–811.

[Gra+00] Grady, M. S. et al. “Experimental study of the magnetic stereotaxis system for catheter manipulation within the brain”. In: *Journal of Neurosurgery* vol. 93, no. 2 (Aug. 2000), pp. 282–288.

[HS97] Hochreiter, S. and Schmidhuber, J. “Long Short-Term Memory”. In: *Neural Computation* vol. 9, no. 8 (Nov. 1997), pp. 1735–1780.

[HWK19] Hyatt, P., Wingate, D., and Killpack, M. D. “Model-Based Control of Soft Actuators Using Learned Non-linear Discrete-Time Models”. In: *Frontiers in Robotics and AI* vol. 6 (2019).

[JGT09] Jones, B. A., Gray, R. L., and Turlapati, K. “Three dimensional statics for continuum robotics”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. ISSN: 2153-0866. Oct. 2009, pp. 2659–2664.

[JPZ14] Jung, J., Penning, R. S., and Zinn, M. R. “A modeling approach for robotic catheters: effects of nonlinear internal device friction”. In: *Advanced Robotics* vol. 28, no. 8 (Apr. 2014), pp. 557–572.

[KH11] Kesner, S. B. and Howe, R. D. “Position Control of Motion Compensation Cardiac Catheters”. In: *IEEE Transactions on Robotics* vol. 27, no. 6 (Dec. 2011), pp. 1045–1055.

[KMR15] Katzschmann, R. K., Marchese, A. D., and Rus, D. “Autonomous Object Manipulation Using a Soft Planar Grasping Manipulator”. In: *Soft Robotics* vol. 2, no. 4 (Dec. 2015), pp. 155–164.

[KS10] Kai Xu and Simaan, N. “Intrinsic wrench estimation and its performance index for multisegment continuum robots”. In: *IEEE Transactions on Robotics* vol. 26, no. 3 (June 2010), pp. 555–561.

[LK90] Lee, H. and Kang, I. S. "Neural algorithm for solving differential equations". In: *Journal of Computational Physics* vol. 91, no. 1 (Nov. 1990), pp. 110–131.

[LLF98] Lagaris, I., Likas, A., and Fotiadis, D. "Artificial neural networks for solving ordinary and partial differential equations". In: *IEEE Transactions on Neural Networks* vol. 9, no. 5 (Sept. 1998), pp. 987–1000.

[Mar+19] Marban, A. et al. "A recurrent convolutional neural network approach for sensorless force estimation in robotic surgery". In: *Biomedical Signal Processing and Control* vol. 50 (Apr. 2019), pp. 134–150.

[McM+06] McMahan, W. et al. "Field trials and testing of the OctArm continuum manipulator". In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* Orlando, FL, USA: IEEE, 2006, pp. 2336–2341.

[MD11] Mahvash, M. and Dupont, P. E. "Stiffness control of surgical continuum manipulators". In: *IEEE Transactions on Robotics* vol. 27, no. 2 (Apr. 2011), pp. 334–345.

[Pai02] Pai, D. K. "Strands: interactive simulation of thin solids using cosserat models". In: *Computer Graphics Forum* vol. 21, no. 3 (Sept. 2002), pp. 347–352.

[PN15] Petruska, A. J. and Nelson, B. J. "Minimum Bounds on the Number of Electromagnets Required for Remote Magnetic Manipulation". In: *IEEE Transactions on Robotics* vol. 31, no. 3 (June 2015), pp. 714–722.

[PU92] Psichogios, D. C. and Ungar, L. H. "A hybrid neural network-first principles approach to process modeling". In: *AIChE Journal* vol. 38, no. 10 (1992), pp. 1499–1511.

[RPK19] Raissi, M., Perdikaris, P., and Karniadakis, G. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* vol. 378 (Feb. 2019), pp. 686–707.

[RW11] Rucker, D. C. and Webster III, R. J. "Statics and Dynamics of Continuum Robots With General Tendon Routing and External Loading". In: *IEEE Transactions on Robotics* vol. 27, no. 6 (Dec. 2011), pp. 1033–1044.

[Sad+19] Sadati, S. et al. "Reduced Order vs. Discretized Lumped System Models with Absolute and Relative States for Continuum Manipulators". In: *Robotics: Science and Systems XV*. Robotics: Science and Systems Foundation, June 2019.

[Sik+17] Sikorski, J. et al. “Introducing BigMag — A novel system for 3D magnetic actuation of flexible surgical manipulators”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. Singapore, Singapore: IEEE, May 2017, pp. 3594–3599.

[Sik+19] Sikorski, J. et al. “Vision-based 3-D control of magnetically actuated catheter using BigMag—An array of mobile electromagnetic coils”. In: *IEEE/ASME Transactions on Mechatronics* vol. 24, no. 2 (2019), pp. 505–516.

[STF04] Simaan, N., Taylor, R., and Flint, P. “A dexterous system for laryngeal surgery”. In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004.* Vol. 1. ISSN: 1050-4729. Apr. 2004, 351–357 Vol.1.

[Tar+20] Tariverdi, A. et al. “Dynamic modeling of soft continuum manipulators using lie group variational integration”. In: *Plos one* vol. 15, no. 7 (2020), e0236121.

[TAR19] Till, J., Aloi, V., and Rucker, C. “Real-time dynamics of soft and continuum robots based on Cosserat rod models”. In: *The International Journal of Robotics Research* vol. 38, no. 6 (May 2019), pp. 723–746.

[Thu+17] Thuruthel, T. G. et al. “Learning dynamic models for open loop predictive control of soft robotic manipulators”. In: *Bioinspiration & Biomimetics* vol. 12, no. 6 (Oct. 2017), p. 066003.

[Thu+19] Thuruthel, T. G. et al. “Soft robot perception using embedded soft sensors and recurrent neural networks”. In: *Science Robotics* vol. 4, no. 26 (Jan. 2019), eaav1488.

[Tur+18] Turan, M. et al. “Deep EndoVO: A recurrent convolutional neural network (RCNN) based visual odometry approach for endoscopic capsule robots”. In: *Neurocomputing* vol. 275 (Jan. 2018), pp. 1861–1870.

[Wen+12] Wen Tang et al. “A stable and real-time nonlinear elastic approach to simulating guidewire and catheter insertions based on cosserat rod”. In: *IEEE Transactions on Biomedical Engineering* vol. 59, no. 8 (Aug. 2012), pp. 2211–2218.

Authors’ addresses

Abbas Tariverdi Department of Physics, University of Oslo, Sem Sælands 24, 0371 Oslo, Norway, abbast@uio.no

Venkatasubramanian K. Venkiteswaran Department of Biomechanical Engineering, University of Twente, 7500 AE Enschede, The Netherlands, v.kalpathyvenkiteswaran@utwente.nl

Michiel Richter Department of Biomechanical Engineering, University of Twente, 7500 AE Enschede, The Netherlands, m.richter@utwente.nl

Ole J. Elle The Intervention Centre, Oslo University Hospital, Oslo, Norway, oelle@ous-hf.no

Jim Tørresen Department of Informatics, University of Oslo, Gaustadalleen 23B, Oslo, Norway, jimtoer@ifi.uio.no

Kim Mathiassen Department of Technology Systems, University of Oslo, Gunnar Randers 19, Oslo, Norway, kim.mathiassen@its.uio.no

Sarthak Misra Department of Biomechanical Engineering, University of Twente, 7500 AE Enschede, The Netherlands, s.misra@utwente.nl

Ørjan G. Martinsen Department of Physics, University of Oslo, Sem Sælands 24, 0371 Oslo, Norway, o.g.martinsen@fys.uio.no

Paper V

Physics-Based Simulation and Control Framework for Steering a Magnetically-Actuated Guidewire

Abbas Tariverdi[1], **Kim Mathiassen, Vegard Søyseth, Håvard Kalvøy, Ole J. Elle, Jim Tørresen, Ørjan G. Martinsen, Mats Erling Høvin**

Published in *8th International Conference on Control, Decision and Information Technologies (CoDIT)*, May 2022, volume 1, pp.770-774. DOI: 10.1109/CoDIT55151.2022.9803963.

Abstract

This paper establishes a physics-based simulation framework for steering a magnetically actuated guidewire based on the linear elasticity and dipoles theories. Interaction wrenches resulting from an external magnetic field and embedded magnets in a continuum rod, i.e., guidewire, serves as actuators for steering. In the presented framework, a simplified integration scheme based on the finite-volume method is employed to model guidewire using the linear elasticity theory and forces resulting from the interference of magnetic fields to provide a rapid model reconstruction. Furthermore, orienting the external magnetic field is employed to steer a guidewire into a constrained environment. Finally, simulations illustrate the approach performance on a soft rod where an external magnetic field is orientated to form the desired shape for a continuum rod and steer it within an environment. The results open up possibilities to construct a rapid model for continuum manipulators in practice.

V

Contents

V.1 Introduction 216
V.2 Problem Statement and Motivation 217
V.3 Method 218
V.4 Simulation Results 222
V.5 Discussion 222

[1]University of Oslo, Postboks 1337 Blindern, 0316 Oslo, Norway, abbast@uio.no

V.6 Conclusion 224
References 224

V.1 Introduction

Modeling and simulation of multiphysics phenomena have attracted the attention of researchers in recent decades with the help of advances in computing performance. The multiphysics simulations by taking into account criteria such as geometrical constraints and mechanical properties. These algorithms can be employed to cope with the difficulties of modeling continuum robots in which various actuation mechanisms are employed.

Soft continuum robots have flexible and stretchable bodies with infinite degrees of freedom, increasing the complexity and maneuverability of robots in a workspace. Modeling these highly complex robots is computationally heavy due to complex geometries and coupled actuation mechanisms such as thermal, electrical, or magnetic actuators. In other words, soft robots may be coupled with other physical fields for actuation purposes, and therefore these physical fields should be taken into account in the modeling.

For continuum manipulators, several actuation mechanisms have been taken into account in the literature such as cable-driven [Ore+16; Qi+21; YL18], thermal [Tel+10], magnetic actuation [Tar+21] in different applications. Table Table V.1 summarizes recent work on shape formation or simulation of continuum manipulators with different actuation mechanisms.

Shape formation of a continuum manipulator can be addressed through model-based [Qi+21; Tar+20] and model-free approaches [Fra+21; Ver+19; YC14]. Although precise modeling of soft robots may lead to a robust control or motion planning, these algorithms [Tar+20] are usually based on numerical methods and target to find solutions for (partial) differential equations and therefore require extensive computational resources. In addition, it may not be feasible to run those algorithms in real-time. On the other side, model-free approaches do not guarantee any level of performance, especially in the presence of uncertainties in the robots' workspace. Therefore, the prevalence of modeling for soft manipulators in real-world applications necessitates developing rapid and intuitive simulations.

This study makes a contribution by presenting a rapid quasi-static modeling technique for simulations of soft manipulators control and formation. This article proposes a rapid multiphysics simulation framework for magnetically actuated continuum manipulators within an environment similar to GI tract for closed-loop control applications to reduce the reality gap. The suggested technique is unique in that it combines rapid quasi-static models with soft guidewires controlled by magnetic fields, which may be employed in closed-loop control systems for precise navigation.

Table V.1: Static simulations of continuum manipulators

Reference	Modeling Approach	Robot
[CWD18]	piecewise-constant curvature	2-DoF tendon-driven robotic guidewire
[Gao+20]	kinematic model considering segments and tension-sensing cable	Cable-Driven Continuum robot with interlocked segments
[Pai02]	Static simulation based on Cosserat rod model in three dimensions	Continuum strands
[GDS19]	Static simulation based on group theory formulation	Intervascular shaping operations
[Tun13]	A finite element based simulation with considering large deformations	A general surgical continuum manipulator
[JGT09]	Static simulations with loads in three dimensions based on Cosserat rod model	A general surgical continuum manipulator
[CCS09]	Static simulation based on linear elasticity theory	A Tendon driven continuum manipulator
[XS08]	Static simulation analysis based on virtual-work and screw theory	Multiple backbone continuum manipulator
[WRC09]	Static simulation based on beam mechanics and elestic energy theory	Concentric tube manipulator

The paper is organized as follows. The problem statement is briefly discussed in Section V.2. Section V.3 is devoted to introducing the approaches we employed for a rapid simulation. Simulations have also been carried out to test the efficiency of the proposed solver in Section V.4, and discussions and conclusions are reported in Section V.5 and Section V.6, respectively.

V.2 Problem Statement and Motivation

Consider a guidewire into which a permanent magnet is embedded as an actuation point. The guidewire will be shaped to go through a Gastrointestinal (GI) tract structure by applying external magnetic fields through single or multiple permanent magnets. Firstly, a fast, controllable model for magnetic wrenches is developed where each permanent magnet is modeled as one or multiple dipoles. Secondly, for the guidewire, multiple joints are considered on the wire where they are modeled as rotational springs (to capture the bending potential energy), and segments in between the joints are modeled as a linear spring. Finally, the interaction of the manipulator and its surrounding tissues, i.e., friction, has been incorporated into the manipulator model. The presented model can be used to

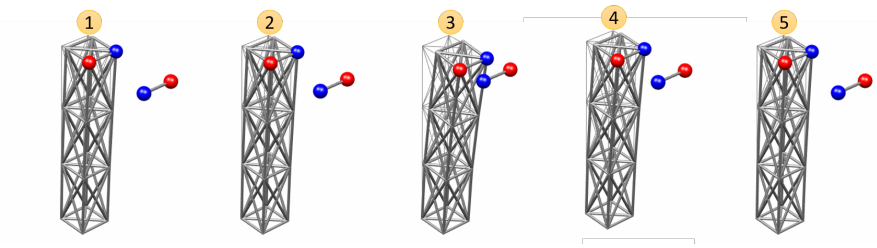


Figure V.1: By moving the free dipole toward the constrained dipole, the beam bends due to the magnetic force resulting from an attraction field.

model both stiff and soft continuum manipulators. In the presented framework, the objective is to steer a guidewire into an intervascular-like structure.

V.3 Method

This section presents a framework for the simulation of a soft continuum manipulator based on the linear elasticity theory.

V.3.1 Magnetic Interaction Modeling

In general, calculating interactive magnetic wrenches between two magnets is a very complex problem, and they depend on the magnetization, shape, and pose of magnets. In other terms, for accurate calculations of the wrenches, numerical methods are employed. Nevertheless, light and real-time models for calculating the wrenches are essential when magnets are used in a control loop as model uncertainties and errors are compensated by designing suitable controllers. In this study, and for simplicity, magnets are represented by dipoles. Forces at position $\bar{d}$ resulting from the interaction of two dipoles with magnetic moments $\bar{\mu}_1$ and $\bar{\mu}_2$ are the same in magnitude with opposite directions can be written as $F = \frac{3\mu_0}{4\pi d^5}\left[\bar{n}_1\bar{\mu}_2 + \bar{n}_2\bar{\mu}_1 + < \bar{\mu}_1, \bar{\mu}_2 > \bar{d} - \frac{5}{d^2}\bar{n}_1\bar{n}_2\bar{d}\right]$ where $\bar{n}_1 =< \bar{\mu}_1, \bar{d} >$, $\bar{n}_2 =< \bar{\mu}_2, \bar{d} >$, $d = \|\bar{d}\|_2$, and $\|\cdot\|_2$ denotes norm 2. The magnetic torque of the dipole $\bar{\mu}_2$ acting on the dipole $\bar{\mu}_1$ is defined as $\bar{\tau}_1 = \bar{\mu}_1 \times \bar{B}_2$. in which $\bar{B}_2$ is the magnetic field of the dipole $\bar{\mu}_2$ and is defines as $\bar{B}_2(\bar{d}) = \frac{\mu_0}{4\pi}\left[\frac{3\bar{d}}{d^5}\bar{n}_2 - \frac{\bar{\mu}_2}{d^3}\right]$. In addition, the magnetic torque of the dipole $\bar{\mu}_1$ acting on the dipole $\bar{\mu}_2$ can be found in a similar way.

The effects of a magnetic force resulting from an attraction field between two aligned dipoles and the magnetic torque resulting from the misalignment of two dipoles are shown in Figure V.1 and Figure V.2, respectively. In other words, by moving the free dipole toward the constrained dipole, the beam bends due to the magnetic force resulting from an attraction field. Furthermore, by rotating the free dipole in place, the beam deflects from its original position due to the magnetic torque resulting from the misalignment of two dipoles.

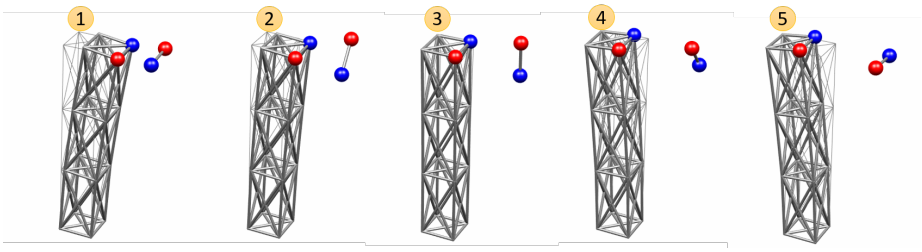


Figure V.2: By rotating the free dipole in place, the beam deflects from its original position due to the magnetic torque resulting from the misalignment of two dipoles.

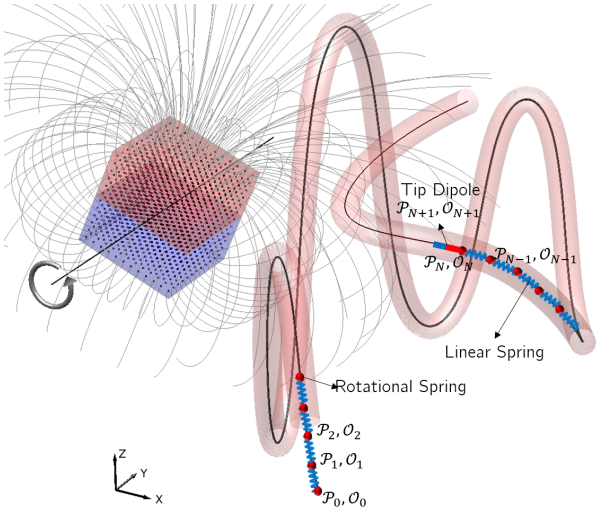


Figure V.3: A guidewire is partitioned into multiple segments and joints: Each segment is modeled as a linear spring and joints are modelled as rotational springs.

V.3.2 Guidewire Modeling

A guidewire is modeled as multiple segments and joints such that each segment is modeled as an extensible spring, and joints are modeled as rotational springs, which allow the guidewire to bend in the three dimensions. The embedded magnet into the tip of the guidewire is considered as a rigid body. Since guidewires are usually thin; therefore, twisting is not considered in the modeling. The model is shown in Figure V.3, and the modeling procedure is summarized in Algorithm 2.

Initialize:
Define a spline curve within the GI tract structure;
Orientation of joints in rest state on spline: $\{\mathcal{O}_0, \cdots, \mathcal{O}_{N+1}\}$;
Position of joints in rest state on spline: $\{\mathcal{P}_0, \cdots, \mathcal{P}_{N+1}\}$;
Initial velocity of joints is zero;
Boundary conditions are considered free at both ends;
while *simulating* **do**

i. Apply external magnetic forces on tip dipole (attached magnet to the tip) alongside with the spline;

ii. Integrate tip magnet position;

iii. Enforce tip magnet position coupling to the rest of the guidewire;

iv. Calculate spring forces between guidewire segments due to segment length deviations by using a spring constant;

v. Calculate bending forces between wire segments (at joints);

vi. Integrate joints poses;

vii. Project all joints forces to the direction of the corresponding guidewire segment (alongside with the spline);

viii. Update joints position;

ix. Snap all joints onto the closest spline;

end

Algorithm 2: Guidewire simulation.

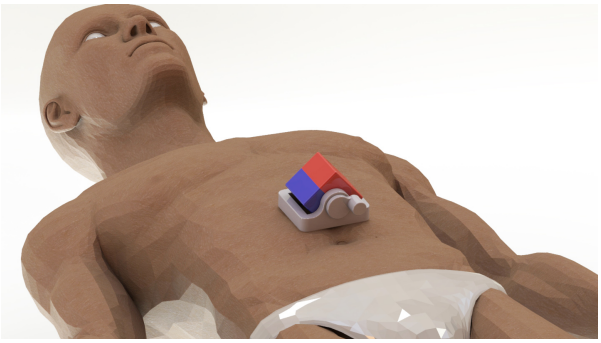


Figure V.4: Proposed steering setup including an external magnet.

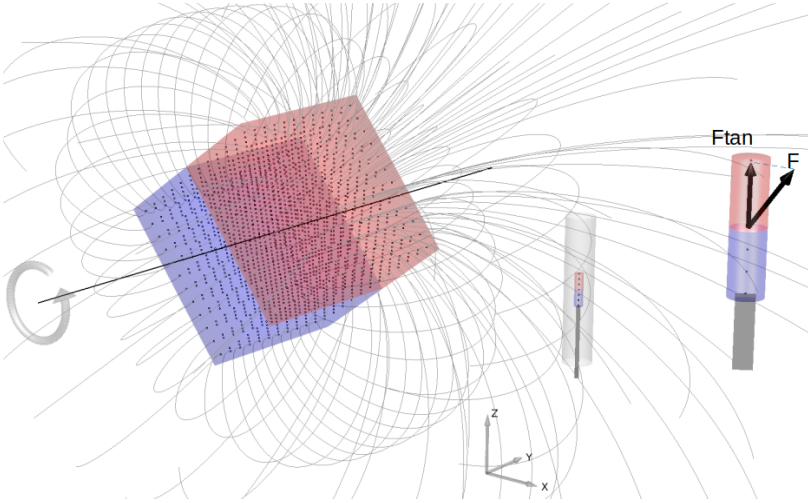


Figure V.5: A rotating external magnet modeled as 2000 dipoles. The resulting flux and the applied force on the tip dipole and the projected force on the spline are shown.

V.3.3 Steering Setup

A single external magnet is employed in this study to introduce forces to the tip for steering purposes. The proposed setup is shown in Figure V.4.

The external magnet is modeled as a set of dipoles (2000 dipoles) rotating around y-axis and Figure V.5 shows the resulting flux and the applied force on the tip dipole and the projected force on the spline.

V.4 Simulation Results

In this section, we consider a permanent magnet modeled as a dipole embedded into the tip of the guidewire. The presented setup, including an external magnet modeled as 2000 dipoles, is located at a 7 cm distance from the GI tract. The idea is to rotate the external magnet to steer the guidewire toward a desired location inside a GI tract structure, as in is shown in Figure V.3. Length and diameter of GI tract structure are 8×10^{-1} m and 1.5×10^{-2} m. The tip embedded neodymium magnet is a cylindrical magnet with a diameter of 2 mm, height 3 mm, weight 7.2×10^{-5} kg, and residual magnetism 1.37 T. A circular cross-section guidewire with the radius 2.5×10^{-3}, Young modulus constants 550 and 600 KPa for rotational and linear springs are considered, i.e., rotational and linear spring constants are 10.79 Nm^{-1} and 11.78 Nm^{-1}, respectively. For the external magnet, a block magnet 50.8 mm $\times$ 50.8 mm $\times$ 50.8 mm with residual magnetism 1.3 T is considered. Furthermore, it should be noted that as boundary conditions, the guidewire is always locked to the direction of the GI tract, i.e., tangent to the GI tract and cannot bungle up inside the environment as it has been mentioned in the last step in Algorithm 1.

For the simulation, the external magnet only has a rotational movement. Figure V.6 shows projected forces tangent to the GI tract structure obtained from different rotation angles. The maximum projected force on the spline curve resulting from a specific rotation angle is depicted in Figure V.7. In other words, Figure V.7 depicts the maximum projected force applied to the dipole embedded at the tip of the guidewire resulting from a specific rotation angle of the external magnet at each position of the GI tract.

V.5 Discussion

It was the primary purpose of the paper to draw attention to the magnetic navigation of a guidewire or catheter inside a complex environment such as the GI tract. The authors attention was concentrated not only on designing a multiphysics simulation framework to model soft guidewires and magnetic fields and forces but also on showing possibilities for applying the proposed framework in minimally invasive surgeries. The originality of the proposed solution lies in the fact that it combines fast quasi-static models of soft guidewires actuated with magnetic fields where this model can be used in closed-loop control systems for accurate navigation.

The results obtained are broadly in good agreement with the major trends of magnetically actuated manipulators. In other words, the main finding in this study is that we only need one magnet and one axis of rotation to move the tip of a guidewire in any direction in 3D space as long as the guidewire tip is aligned with (tangent to) the GI tract. This is not very intuitive that with only one input parameter, the system is capable of dragging a guidewire in the 3D space.

The findings have a number of possible limitations, namely, force magnitude inserted on intestines or the force magnitude might not be enough for moving

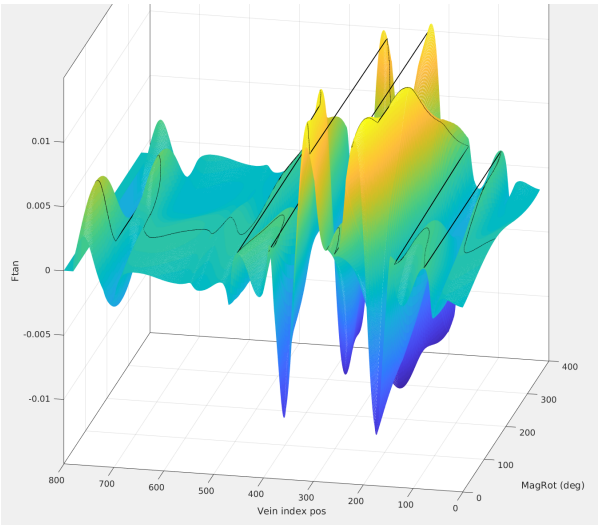


Figure V.6: Projected force (N) applied to the guidewire tip dipole alongside the GI tract resulting from orientation of the external magnet with respect to the rotation of the external magnet at different locations of the GI tract i.e., the GI tract index positions. Maximum projected force applied to the dipole embedded at the tip of the guidewire is shown by the solid black curve.

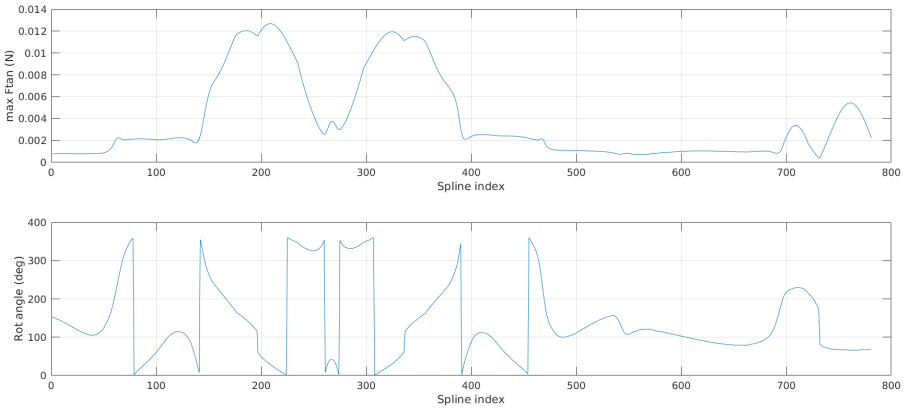


Figure V.7: Maximum projected force applied to the dipole embedded at the tip of the guidewire resulting from a specific rotation angle of the external magnet at each position of the GI tract.

the tip magnet; finally, geometry and the number of the external magnets affect the produced forces on the tip. Also, torques on the tip are not desired in the considered scenario, meaning that torques can deviate from the tip for the defined spline curve. Therefore, by adding more degrees of freedom to the system, such as increasing the number of external magnets and considering transnational movements of external magnets, the magnitude of the projected force can be increased, and undesired effects of torques can be weakened.

V.6 Conclusion

The article demonstrates the possibility of steering a guidewire by rotating an external magnet inside a complex environment. The proposed simulation framework results should be applicable also to different scenarios where the number of external magnets and their geometries are customized. Also, the results open up possibilities to construct a rapid model for magnetically actuated continuum manipulators for closed-loop control applications to reduce the reality gap. Clearly, further research will be needed to validate the results. Therefore, the next stage of this research will be the experimental confirmation of the presented framework. Future work will involve considering more external magnets as actuators together with difficult-to-reach environments.

References

[CCS09] Camarillo, D., Carlson, C., and Salisbury, J. "Configuration tracking for continuum manipulators with coupled tendon drive". In: *IEEE Transactions on Robotics* vol. 25, no. 4 (Aug. 2009), pp. 798–808.

[CWD18] Chitalia, Y., Wang, X., and Desai, J. P. "Design, modeling and control of a 2-dof robotic guidewire". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 32–37.

[Fra+21] Frazelle, C. et al. "Kinematic-Model-Free Control for Space Operations with Continuum Manipulators". In: *2021 IEEE Aerospace Conference (50100)*. Big Sky, MT, USA: IEEE, Mar. 2021, pp. 1–11.

[Gao+20] Gao, A. et al. "Laser-profiled continuum robot with integrated tension sensing for simultaneous shape and tip force estimation". In: *Soft Robotics* vol. 7, no. 4 (2020), pp. 421–443.

[GDS19] Grazioso, S., Di Gironimo, G., and Siciliano, B. "A geometrically exact model for soft continuum robots: The finite element deformation space formulation". In: *Soft robotics* vol. 6, no. 6 (2019), pp. 790–811.

[JGT09] Jones, B. A., Gray, R. L., and Turlapati, K. "Three dimensional statics for continuum robotics". In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. ISSN: 2153-0866. Oct. 2009, pp. 2659–2664.

[Ore+16] Orekhov, A. L. et al. “Analysis and validation of a teleoperated surgical parallel continuum manipulator”. In: *IEEE Robotics and Automation Letters* vol. 1, no. 2 (2016), pp. 828–835.

[Pai02] Pai, D. K. “Strands: interactive simulation of thin solids using cosserat models”. en. In: *Computer Graphics Forum* vol. 21, no. 3 (Sept. 2002), pp. 347–352.

[Qi+21] Qi, F. et al. “Dynamic model and control for a cable-driven continuum manipulator used for minimally invasive surgery”. In: *The International Journal of Medical Robotics and Computer Assisted Surgery* (2021).

[Tar+20] Tariverdi, A. et al. “Dynamic modeling of soft continuum manipulators using lie group variational integration”. In: *Plos one* vol. 15, no. 7 (2020), e0236121.

[Tar+21] Tariverdi, A. et al. “A Recurrent Neural-Network-Based Real-Time Dynamic Model for Soft Continuum Manipulators”. In: *Frontiers in Robotics and AI* vol. 8 (2021), p. 45.

[Tel+10] Telleria, M. J. et al. “Modeling and implementation of solder-activated joints for single-actuator, centimeter-scale robotic mechanisms”. In: *2010 IEEE International Conference on Robotics and Automation*. IEEE. 2010, pp. 1681–1686.

[Tun13] Tunay, I. “Spatial continuum models of rods undergoing large deformation and inflation”. In: *IEEE Transactions on Robotics* vol. 29, no. 2 (Apr. 2013), pp. 297–307.

[Ver+19] Verghese, M. et al. “Model-free Visual Control for Continuum Robot Manipulators via Orientation Adaptation”. In: *arXiv:1909.00450 [cs]* (Sept. 2019). arXiv: 1909.00450.

[WRC09] Webster, R., Romano, J., and Cowan, N. “Mechanics of precurved-tube continuum robots”. In: *IEEE Transactions on Robotics* vol. 25, no. 1 (Feb. 2009), pp. 67–78.

[XS08] Xu, K. and Simaan, N. “An investigation of the intrinsic force sensing capabilities of continuum robots”. In: *IEEE Transactions on Robotics* vol. 24, no. 3 (June 2008), pp. 576–587.

[YC14] Yip, M. C. and Camarillo, D. B. “Model-Less Feedback Control of Continuum Manipulators in Constrained Environments”. In: *IEEE Transactions on Robotics* vol. 30, no. 4 (Aug. 2014), pp. 880–889.

[YL18] Yuan, H. and Li, Z. “Workspace analysis of cable-driven continuum manipulators based on static model”. In: *Robotics and Computer-Integrated Manufacturing* vol. 49 (2018), pp. 240–252.

Authors’ addresses

Abbas Tariverdi Department of Physics, University of Oslo, Sem Sælands 24, 0371 Oslo, Norway, abbast@uio.no

Kim Mathiassen Department of Technology Systems, University of Oslo, Gunnar Randers 19, Oslo, Norway, kim.mathiassen@its.uio.no

Vegard Søyseth Department of Informatics, University of Oslo, Gaustadalleen 23B, Oslo, Norway, vegardds@ifi.uio.no

Håvard Kalvøy Department of Clinical and Biomedical Engineering, Oslo University Hospital, Oslo, Norway, havard.kalvoy@ous-hf.no

Ole J. Elle The Intervention Centre, Oslo University Hospital, Oslo, Norway, oelle@ous-hf.no

Jim Tørresen Department of Informatics, University of Oslo, Gaustadalleen 23B, Oslo, Norway, jimtoer@ifi.uio.no

Ørjan G. Martinsen Department of Physics, University of Oslo, Sem Sælands 24, 0371 Oslo, Norway, o.g.martinsen@fys.uio.no

Mats Erling Høvin Department of Informatics, University of Oslo, Gaustadalleen 23B, Oslo, Norway, matsh@ifi.uio.no

Appendices

Appendix A

Some Mathematical Concepts with Physical Interpretations

Symplectic Systems [1]

Symplectic systems are a class of mathematical systems that describe the behavior of many physical phenomena, particularly in classical mechanics. They provide a framework for understanding how certain quantities evolve over time, such as position and momentum. To give an intuitive and cool interpretation of symplectic systems, let's consider an analogy involving a dance.

Imagine a dance floor filled with couples, where each couple represents a pair of canonical coordinates (position and momentum) of a particle in a physical system. The symplectic structure can be thought of as the "choreography" that guides the couples through the dance, preserving certain relationships between their movements.

Now, imagine that the dance floor is a phase space, a mathematical space where each point represents the complete state of the system (i.e., the positions and momenta of all particles). The symplectic structure helps us understand how the system evolves over time, tracing out paths through phase space as the particles move.

The key aspect of a symplectic system is the conservation of a certain geometric property called the symplectic form, which is analogous to the way couples maintain constant distance and orientation during a dance. This conservation principle has important implications for the behavior of physical systems:

- Conservation of energy: Just as dancers must follow certain rules to maintain their rhythm and balance, symplectic systems must conserve energy. This principle is crucial for understanding the long-term behavior of physical systems.

- Reversibility: Imagine playing a video of the dance in reverse. The symplectic structure ensures that the dance still "works" and follows the same choreography, just in the opposite direction. Similarly, symplectic systems exhibit time-reversibility, meaning that if you reverse the direction of time, the equations of motion still hold.

- Stability: The conservation of the symplectic form also helps maintain stability in the system. Just as the rules of the dance prevent couples from

[1] [Arn89; Can01; GS90; MS98]

colliding or drifting apart, the symplectic structure prevents the system from becoming chaotic or unstable.

Overall, symplectic systems provide a powerful and elegant way to describe the dynamics of many physical systems, with the symplectic structure playing the role of an intricate and well-orchestrated dance that maintains the harmony and balance of the system.

$SO(2)$, $SE(2)$, $SO(3)$, and $SE(3)$ [2]

- $SO(n)$: Special Orthogonal group of order n, represents rotations in n-dimensional space.
- $SE(n)$: Special Euclidean group of order n, represents rigid body motions (rotations and translations) in $n-$dimensional space.
- $so(n)$: The Lie algebra of the Special Orthogonal group $SO(n)$, represents infinitesimal rotations in $n-$dimensional space.
- $se(n)$: The Lie algebra of the Special Euclidean group $SE(n)$, represents infinitesimal rigid body motions (rotations and translations) in $n-$dimensional space.

Now let's dive into each one with examples in 2D and 3D space:

- $SO(2)$ - Rotations in 2D space: Imagine you have a flat piece of paper with a drawing of an arrow. $SO(2)$ represents all possible rotations of that arrow around a fixed point (like the arrow's tail) without flipping or deforming it. This can be visualized as a continuous rotation from 0 to 360 degrees.
- $SE(2)$ - Rigid body motions in 2D space: Now, in addition to rotating the arrow on the paper, you can also slide it around without changing its orientation. $SE(2)$ represents all possible combinations of rotations and translations (sliding) of the arrow in the 2D plane. This can be visualized as moving the arrow to any position on the paper while also allowing it to rotate.
- $SO(3)$ - Rotations in 3D space: Imagine a small 3D arrow model suspended in mid-air. $SO(3)$ represents all possible rotations of that arrow around any axis in 3D space. You can rotate the arrow around the $x-$, $y-$, and $z-$axes, or any combination of these axes. This can be visualized as a continuous rotation around any axis from 0 to 360 degrees.

[2] [Lip66; ŠŠ20; Yos90]

- $SE(3)$ - Rigid body motions in 3D space: In addition to rotating the 3D arrow, you can also move it around in the 3D space without changing its orientation. $SE(3)$ represents all possible combinations of rotations and translations of the arrow in 3D space. This can be visualized as moving the arrow to any position in 3D space while also allowing it to rotate around any axis.

- $so(2)$ - Infinitesimal rotations in 2D space: Going back to the 2D arrow on the paper, $so(2)$ represents the tangent space of $SO(2)$ at the identity element. This captures the infinitesimal rotations of the arrow, i.e., extremely small rotations, around the fixed point. You can think of this as the "speed" or "rate" of rotation of the arrow.

- $se(2)$ - Infinitesimal rigid body motions in 2D space: se(2) represents the tangent space of $SE(2)$ at the identity element, which captures the infinitesimal rigid body motions of the arrow in 2D space. This includes both small rotations and small translations (sliding) of the arrow. You can think of this as the "speed" or "rate" of rotation and translation of the arrow in the 2D plane.

- $so(3)$ - Infinitesimal rotations in 3D space: For the 3D arrow model suspended in mid-air, $so(3)$ represents the tangent space of $SO(3)$ at the identity element. This captures the infinitesimal rotations of the arrow in 3D space, i.e., extremely small rotations, around any axis. You can think of this as the "speed" or "rate" of rotation of the arrow around any axis in 3D space.

- $se(3)$ - Infinitesimal rigid body motions in 3D space: se(3) represents the tangent space of $SE(3)$ at the identity element, which captures the infinitesimal rigid body motions of the arrow in 3D space. This includes both small rotations and small translations of the arrow. You can think of this as the "speed" or "rate" of rotation and translation of the arrow in 3D space.

In summary, $SO(n)$ and $SE(n)$ are groups that represent rotations and rigid body motions in $n-$dimensional space, respectively. $so(n)$ and $se(n)$ are their corresponding Lie algebras, representing infinitesimal rotations and rigid body motions. These concepts are fundamental in various fields such as robotics, computer graphics, and physics, as they help describe the motion and transformations of objects in space in a mathematically consistent and compact manner.

Lie groups and Lie algebras [3]

Consider a flat, 2D plane with an equilateral triangle on it. Now, think of all the possible ways to move this triangle on the plane without changing its shape

[3] [Gil08; Hal15]

or orientation. These transformations include translations (sliding the triangle) and rotations (spinning the triangle around a fixed point). We can describe this set of transformations as a Lie group. The translations form a Lie group called $\mathbb{R}^2$, representing translations along the x and y axes. The rotations around a fixed point form another Lie group called $SO(2)$, which describes rotations in a 2D plane.

Now, let's consider a Rubik's Cube. The set of all possible rotations of the cube's faces forms a discrete group called the Rubik's Cube group. However, if we allow smooth, continuous rotations of the faces, we obtain a Lie group called $SO(3)$ that describes rotations in 3D space.

To understand Lie algebras, let's go back to the 2D plane with the equilateral triangle. When we perform a small rotation or translation, we're essentially moving along a tangent direction to the identity transformation (i.e., no transformation). The set of all such tangent directions forms the Lie algebra associated with the Lie group. For the $\mathbb{R}^2$ Lie group, the Lie algebra is also denoted as $\mathbb{R}^2$. It consists of two generators: one for translations along the $x-$axis and one for translations along the $y-$axis. The Lie algebra for $SO(2)$ consists of infinitesimal rotations in the plane, with a single generator corresponding to the rotation around the fixed point.

In the Rubik's Cube example, the associated Lie algebra for the continuous $SO(3)$ group consists of infinitesimal rotations around the $x-$, $y-$, and $z-$ axes. Each axis has a generator, and their linear combinations describe all possible infinitesimal rotations in 3D space.

Every Lie group has an associated Lie algebra, which can be thought of as capturing the local behavior of the group. Conversely, every Lie algebra can be "integrated" to obtain the corresponding Lie group. This integration process is called exponentiation. In physics, Lie groups and Lie algebras often appear in tandem, with the Lie group describing the global symmetries and the Lie algebra describing the generators of these symmetries.

Lie groups have both algebraic and topological properties, while Lie algebras are purely algebraic structures. Lie groups describe global symmetries, while Lie algebras describe local (infinitesimal) symmetries. Lie groups are composed of transformations (e.g., rotations or translations), while Lie algebras consist of the tangent space at the identity element of the Lie group.

In summary, Lie groups describe global, continuous symmetries (such as translations and rotations), while Lie algebras capture the infinitesimal, local structure of those symmetries.

References

[Arn89] Arnold, V. *Mathematical Methods of Classical Mechanics*. New York: Springer-Verlag, 1989.

[Can01] Cannas da Silva, A. *Lectures on Symplectic Geometry*. Berlin: Springer-Verlag, 2001.

[Gil08] Gilmore, R. *Lie Groups, Physics, and Geometry: An Introduction for Physicists, Engineers and Chemists*. Cambridge University Press, 2008.

[GS90] Guillemin, V. and Sternberg, S. *Symplectic Techniques in Physics*. Cambridge: Cambridge University Press, 1990.

[Hal15] Hall, B. C. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Springer, 2015.

[Lip66] Lipkin, H. J. *Lie Groups for Pedestrians*. Amsterdam: North-Holland, 1966.

[MS98] McDuff, D. and Salamon, D. *Introduction to Symplectic Topology*. Oxford: Oxford University Press, 1998.

[ŠŠ20] Šilhan, K. and Šilhan, J. *Geometry of Lie Groups*. Cham: Springer, 2020.

[Yos90] Yoshikawa, T. *Foundations of Robotics: Analysis and Control*. Cambridge: MIT Press, 1990.