LEARNING FROM DEMONSTRATION AND ADAPTIVE SHARED CONTROL FOR EFFECTIVE HUMAN-ROBOT COLLABORATION

A Dissertation

by

Negin Amirshirzad

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in the
Department of Computer Science

Özyeğin University
October 2024

Copyright © 2024 by Negin Amirshirzad

LEARNING FROM DEMONSTRATION AND ADAPTIVE SHARED CONTROL FOR EFFECTIVE HUMAN-ROBOT COLLABORATION

Approved by:

Prof. Erhan Öztop, Advisor
Department of Artificial Intelligence and Data Engineering
Özyeğin University

Asst. Prof. İlknur Karadeniz
Department of Artificial Intelligence and Data Engineering
Özyeğin University

Assoc. Prof. Emre Uğur
Department of Computer Engineering
Boğaziçi University

Asst. Prof. İnci M. Baytaş
Department of Computer Engineering
Boğaziçi University

Date Approved: 12 August 2024

Assoc. Prof. Özkan Bebek
Department of Mechanical Engineering
Özyeğin University

To my parents, Azim and Farideh, whose endless patience, support and encouraging phone calls made all the difference. Your words of encouragement were as crucial as the caffeine and comfort food that powered me through those late-night exam marathons.

To my siblings, Nooshin and Mohammad, whose occasional teasing and unwavering support kept me grounded and reminded me that family is the best source of inspiration and reality check.

ABSTRACT

This thesis explores human-robot shared control systems and the use of learning from demonstration methods to transfer skills from humans to robots. The initial investigation focuses on human adaptation to these systems, revealing that shared control helps humans learn tasks faster and adapt more effectively to robot behavior. Building on these findings, an adaptive blended human-robot shared control framework is proposed and assessed, emphasizing the importance of defining the scope of knowledge for the robot to accurately estimate human intention. This framework dynamically adjusts control weights based on the confidence of this estimate, optimizing the interaction between human and robot. Real-time estimation of human intentions allows the robot to adjust its role, whether passive, leading, or collaborative. The framework is applied to a surgical suturing task, demonstrating its effectiveness in improving performance. By combining human precision with robotic support, the system enhances both efficiency and accuracy, underscoring the effectiveness of adaptive control systems that integrate human expertise with autonomous capabilities.

Additionally, Context-based Echo State Networks (CESNs) are introduced as a lightweight solution for generating movement primitives in robotics. CESNs leverage context vectors to modulate reservoir dynamics, producing varied movement patterns with a single linear read-out weight vector. CESNs' computational efficiency, minimal training requirements, and resilience to noise and delays make them a promising learning from demonstration method for robotic applications.

ÖZETÇE

Bu tez, insan-robot paylaşımlı kontrol sistemlerini ve insanlardan robotlara beceri aktarımını sağlamak için gösterimden öğrenme yöntemlerinin kullanımını araştırmaktadır. İlk inceleme, bu sistemlere insan uyumuna odaklanarak, paylaşımlı kontrolün insanların görevleri daha hızlı öğrenmesine ve robot davranışına daha etkili uyum sağlamasına yardımcı olduğunu ortaya koymaktadır. Bu bulgular üzerine inşa edilen uyarlanabilir bir harmanlanmış insan-robot paylaşımlı kontrol çerçevesi önerilmiş ve değerlendirilmiştir. Çalışma robotun insan niyetini doğru bir şekilde tahmin etmesi için bilgi kapsamının tanımlanmasının önemini vurgulamaktadır. Bu çerçeve, bu tahminin güvenilirliğine dayanarak kontrol ağırlıklarını dinamik olarak ayarlayarak insan ve robot etkileşimini optimize eder. Geliştirilen sistem insan niyetlerinin gerçek zamanlı tahmini, robotun pasif, lider veya işbirlikçi olsun, rolünü ayarlamasına olanak tanır. Çerçeve, cerrahi dikiş atma görevine uygulanmış ve performansı artırmadaki etkinliğini göstermiştir. İnsan hassasiyeti ile robotik desteği birleştirerek, sistem hem verimliliği hem de doğruluğu artırır ve insan uzmanlığı ile otonom yetenekleri entegre eden uyarlanabilir kontrol sistemlerinin etkinliğini vurgular.

Ek olarak, bu tezde bağlam Tabanlı Yankı Durum Ağları (CESN'ler), robotikte hareket primitifleri üretmek için hafif bir çözüm olarak geliştirilmiştir. CESN'ler, rezervuar dinamiklerini modüle etmek için bağlam vektörlerinden yararlanarak tek bir doğrusal okuma ağırlık vektörü ile çeşitli hareket kalıpları üretir. CESN'lerin hesaplama verimliliği, minimal eğitim gereksinimleri ve gürültü ve gecikmelere karşı dayanıklılığı, onları robotik uygulamalar için gösterimden öğrenme yöntemi olarak umut verici kılmaktadır.

ACKNOWLEDGEMENTS

I extend my deepest gratitude to Professor Erhan Oztop, my mentor and advisor for the past decade. His guidance has been invaluable, teaching me the art of research, effective problem-solving, time management, and perseverance. Professor Oztop's ability to blend professionalism with genuine care has been a cornerstone of my academic journey. His support and the numerous opportunities he has provided have been crucial in shaping my research skills and professional growth. He has truly been my ultimate academic GPS (Grand Professor of Science).

I am also thankful to Dr. Ozkan Bebek for his expertise in navigating the complexities of the research laboratory and for his effective leadership and practical advice, which have greatly enriched my experience.

My sincere appreciation goes to all my jury members for their constructive feedback and insightful suggestions throughout this process.

Special thanks to Professor Hasan Sözer, the Chair of the Computer Science Department, Professor Tanju Erdem, the Vice Rector, and Professor Fatih Uğurdağ, the Dean of the Engineering Faculty, for their unwavering support and leadership throughout my academic journey.

The engineering faculty staff has been instrumental in assisting and overcoming bureaucratic hurdles, making the administrative aspects of my journey much smoother.

Özyeğin University deserves recognition for fostering a dynamic and engaging environment that made the academic experience both enjoyable and rewarding.

This research was supported by the Scientific and Technological Research Council of Türkiye (TÜBİTAK), under grant number 119E036.

TABLE OF CONTENTS

DEDICATION . . . . . **iii**

ABSTRACT . . . . . **iv**

ÖZETÇE . . . . . **v**

ACKNOWLEDGEMENTS . . . . . **vi**

LIST OF TABLES . . . . . **xii**

LIST OF FIGURES . . . . . **xiii**

I INTRODUCTION **3**

1.1 Background 3

1.2 Research Problem 4

1.3 Objectives and Aims 5

1.4 Significance of the Study 6

1.5 Limitations of the Research 8

II LITERATURE REVIEW **10**

2.1 Human-Robot Interaction 10

2.2 Adaptive Robotics 11

2.3 Shared Control 12

2.4 Adaptive Shared Control 14

2.4.1 Applications of Adaptive Shared Control 14

2.5 Human Intention Estimation 15

2.6 Learning from Demonstration 17

2.7 Interface Design for Human-Robot Systems 18

2.8 Human to Robot Skill Transfer 19

2.9 Challenges in Adaptive Human-Robot Shared Control 24

III HUMAN ADAPTATION TO HUMAN-ROBOT SYSTEMS . . . **26**

3.1 Shared Control Framework 27

3.2 Designed Task 28
3.3 Robotic Setup 28
3.3.1 Robot Teleoperation 28
3.3.2 Autonomous Controller 29
3.3.2.1 Human Intention Prediction 30
3.3.3 Shared Controller 32
3.4 Experiments 32
3.4.1 Participants 33
3.4.2 Experiment Design 33
3.4.3 Performance Measure 34
3.4.4 Analysis Method 34
3.5 Results 34
3.5.1 Performance Results 34
3.5.1.1 Interpretation 36
3.5.2 Learning Progress Results 37
3.5.2.1 Learning Rate 40
3.5.3 Survey 41
3.5.4 Discussion 42
3.6 Conclusion 43
IV ADAPTIVE BLENDED HUMAN-ROBOT SHARED CONTROL 44
4.1 Introduction 44
4.2 Adaptive Blended Shared Control Framework 47
4.2.1 Human Control 47
4.2.1.1 Feedforward Interface 48
4.2.1.2 Feedback Interface 48
4.2.2 Robot Control 49
4.2.2.1 Human Intention Estimator 49
4.2.2.2 Scope of Knowledge 49

4.2.2.3 Intention to State 49
4.2.3 Shared Control 50
4.2.3.1 Control Command Generator 50
4.2.3.2 Blending Parameter Estimator 50
4.2.3.3 Shared Command Generator 51
4.3 Human to Robot Skill Transfer 51
4.3.1 Learning from weighted demonstrations 52
4.4 Human Intention Estimation 55
4.4.1 Adaptive shared control policy based on human intention . . 55
4.5 Framework Assessment 57
4.5.1 Human intention estimation assessment 57
4.5.1.1 Teaching shared trajectories to robot 60
4.5.2 Scope of Knowledge Assessment 62
4.5.2.1 Teleoperation 63
4.5.2.2 Collecting Demonstrations 63
4.5.2.3 Skill transfer to robot 64
4.5.2.4 Learning Scope of Knowledge 64
4.5.3 User Study 65
4.6 Discussion 69
4.7 Conclusion 70
V SHARED CONTROL IN MEDICAL SUTURING 71
5.1 Introduction 71
5.2 Shared Control for Surgical Needle Insertion 74
5.2.1 Teleoperation 76
5.2.2 Collecting Demonstrations 77
5.2.3 Autonomous Skill Acquisition 78
5.2.4 Graphical User Interface 78
5.2.4.1 Learning Scope of Knowledge 81

5.3 Experiment Design 82
5.4 Results 85
5.5 Discussion 86
5.6 Conclusion 87
5.7 Future Work Directions 88

VI RESERVOIR COMPUTING FOR LEARNING FROM DEMONSTRATION AND SHARED CONTROL 90

6.1 Introduction 90
6.2 Echo State Networks 92
6.3 Context-based Echo State Networks 93
6.3.1 Data Augmentation with Noise 95
6.3.2 CESN Implementation 95
6.4 Experiments and Results 96
6.4.1 2D Reaching with No Feedback 96
6.4.1.1 Experiment Design 96
6.4.1.2 Results 96
6.4.2 3D Reaching Task 97
6.4.2.1 Experiment Design 97
6.4.2.2 Results 99
6.4.3 Obstacle Avoidance Task 100
6.4.3.1 Experiment Design 104
6.4.3.2 Results 104
6.4.4 Discussion 105
6.5 CESNs with Prediction Confidence for Human-Robot Shared Control 107
6.5.1 Prediction Interval 108
6.6 Emulating CNMPs with CESNs 110
6.6.1 Single context point 110
6.6.2 Multiple context points 113
6.7 Conclusion 114

VII CONCLUSION AND FUTURE WORK 116

7.1 Summary of Findings 116

7.2 Future Work 117

VIII MY EPILOGUE 119

REFERENCES 120

LIST OF TABLES

1 Statistical analysis results of shared and human control conditions over different days. 36

2 R-Squared Value of the fitted regression functions comparison of Linear, Exponential, and Power fit for Shared Control and Human Control conditions. 37

3 Learning rates of subjects under different control conditions 40

4 Learning rate comparison 41

5 Comparison between conditions for reaching with obstacle avoidance task 67

6 Descriptive statistics of within-subject factors 68

7 Repeated measures multivariate analysis of variance 68

8 Surgical needle insertion task - Error of autonomous synthesis of wrist trajectory 80

9 Surgical needle insertion task - Error of autonomous synthesis of hand trajectory 80

10 Descriptive statics of performance metrics in the successful trials in both setups for surgical needle insertion task 86

11 Accuracy of the 2D reaching task using the Franka Emika robot for different target points given as context to CESN. 97

12 Accuracy of the 3D reaching task using the Torobo humanoid robot for different target points given as context to CESN. 100

LIST OF FIGURES

1 Conditional neural processes architecture. Image credit: [1] 21
2 Conditional neural movement primitives. Image credit: [2] 23
3 Ball balancing task robotic setup 29
4 Actual and predicted commands obtained through linear regression . 31
5 Daily performance of human control and shared control 35
6 Fitted power function on the performance of shared control group . . 38
7 Fitted power function on the performance of human control group . . 39
8 Learning rate of shared and human control condition 40
9 Adaptive blended human-robot shared control 47
10 Weighted-CNMP vs CNMP generated trajectory 54
11 Mobile robot simulation scene 57
12 Mobile robot trajectory and the intention estimation 60
13 Mobile robot shared parameters 61
14 Blending parameter and difference between human and robot commands 62
15 Simulated scene for the obstacle avoidance task in Coppeliasim 63
16 Trajectory synthesis by Weighted-CNMP 64
17 Representation of robot's scope of knowledge 66
18 Surgical needle insertion task environment 75
19 Surgical needle insertion task objective 75
20 Data flow of the teleoperation system 77
21 Surgical needle insertion task 79
22 Graphical user interface 81
23 Representation of robot's scope of knowledge 83
24 Context based Echo State Network architecture 94
25 2D reaching task environment 98
26 Effect of data augmentation with noise 101
27 Data augmentation and CESN trajectory estimation 102

28 3d reaching task environment 103
29 Torobo performance in reaching different target points 103
30 Obstacle avoidance task environment 105
31 Torobo performance in the obstacle avoidance task 106
32 CNMP prediction vs CESN prediction 111
33 Comparison of estimated trajectories and confidence levels between CNMP and CESN models - single context 112
34 Comparison of estimated trajectories and confidence levels between CNMP and CESN models - multi context 113

Abbreviations and Acronyms

AI Artificial Intelligence

ANNs Artificial Neural Networks

ANPs Attentive Neural Processes

CESNs Context based Echo State Networks

CNPs Conditional Neural Processes

DMPs Dynamic Movement Primitives

ESNs Echo State Networks

GMR Gaussian Mixture Regression

GPR Gaussian Processes Regression

GUI Graphic User Interface

HRC Human-Robot Collaboration

HRI Human-Robot Interaction

LfD Learning from Demonstration

MLP Multi Layer Perceptron

MPs Movement Primitives

MSE Mean Squared Error

NPs Neural Processes

ProMPs Probabilistic Movement Primitives

ReLU Rectified Linear Unit

RL Reinforcement Learning

RNNs Recurrent Neural Networks

ROS Robot Operating System

SC Shared Control

SDTV Standard Deviation

SEM Standard Error of the Mean

CHAPTER I

INTRODUCTION

1.1 Background

The field of robotics and artificial intelligence is rapidly transforming, particularly in enhancing human-robot interaction (HRI) systems. These interactive systems are designed to facilitate collaboration between humans and machines, leveraging the strengths of both to perform tasks more efficiently and effectively. One emerging area within HRI is adaptive human-robot shared control, where control over a task is dynamically shared between the human operator and the robotic system. This approach aims to combine human cognitive and decision-making capabilities with the precision and endurance of robots [3].

The concept of shared control is not entirely new; it has been explored in various applications such as teleoperation, assistive technologies, and autonomous vehicles [4]. However, traditional models often rely on predefined rules and static allocation of control, which are no longer efficient for complex and dynamic environments. Adaptive shared control seeks to address these limitations by continuously adjusting the level of control based on the real-time assessment of the human operator's performance, intent and physiological state, as well as the environmental context and task requirements [5].

The need for adaptive shared control systems is particularly important in scenarios where the tasks are too complex or hazardous for a human to handle alone, but where full automation is not yet feasible or desirable due to the need for human judgment and adaptability. Examples of such scenarios include robotic surgeries, where precision and control are critical [3], and industrial robotics, where human

expertise and oversight are essential to handle unexpected situations [6].

Research in adaptive human-robot shared control draws from various disciplines, including robotics, artificial intelligence, human factors engineering, and cognitive science. It involves developing algorithms for real-time data processing and decision-making, designing intuitive interfaces for seamless interaction, and conducting rigorous evaluations to ensure safety, reliability, and user satisfaction [7, 8].

This thesis aims to advance the field of adaptive human-robot shared control by investigating new methodologies for dynamic control allocation and developing robust systems that can operate effectively in a variety of real-world applications. By improving the synergy between humans and machines, this research seeks to create more versatile, efficient, and safe collaborative systems, ultimately contributing to the broader goal of integrating adaptive robotics into everyday human activities and industries [9, 10].

1.2 Research Problem

Despite advances in human-robot interaction and shared control systems, several challenges remain in developing adaptive shared control mechanisms that are both efficient and user-friendly. The primary research question guiding this thesis is:

How can adaptive shared control systems be designed to dynamically allocate control between human operators and robotic systems in a way that maximizes task performance, enhances safety, and improves user experience in complex and dynamic environments?

This overarching question can be broken down into several sub-questions:

- What are the key factors influencing the effectiveness of control allocation between humans and robots in various tasks?
- How can real-time assessment of human operator performance and intent be integrated into control algorithms?

- What interface designs best facilitate intuitive and seamless interaction between human operators and robotic systems?
- How can adaptive shared control systems be validated and tested to ensure they meet safety and reliability standards?

Addressing these questions involves a multidisciplinary approach that combines insights from robotics, artificial intelligence, and human factors engineering. The outcomes of this research aim to provide a framework for developing adaptive shared control systems that are versatile, efficient, and safe, thereby enhancing human-robot collaboration in a wide range of applications.

1.3 Objectives and Aims

The main objective of this thesis is to develop adaptive shared control systems that dynamically allocate control between human operators and robotic systems to enhance task performance, safety, and user experience, while decreasing the human effort in complex and dynamic environments. To achieve this goal, the following goals are defined:

1. **Develop Learning from Demonstration Methods:** Explore and implement learning from demonstration (LfD) techniques to transfer skills from human operators to robotic systems. This involves capturing and analyzing human demonstrations to enable robots to learn and replicate complex tasks, thereby enhancing their autonomy and effectiveness.
2. **Develop Algorithms for Dynamic Control Allocation:** Design and implement algorithms that can dynamically adjust the level of control between the human operator and the robot based on real-time data. This includes assessing the operator's performance, and intention, as well as the environmental context and task requirements.

3. **Create Intuitive Human-Robot Interfaces:** Design user interfaces that facilitate seamless interaction between human operators and robotic systems. These interfaces should be intuitive, reducing the cognitive load on the operator and enabling efficient control.

4. **Evaluate System Performance and Safety:** Conduct testing and evaluation of the developed systems to ensure they meet standards of performance and safety. This involves both simulation studies and real-world experiments to validate the effectiveness and reliability of the adaptive shared control systems.

5. **Enhance User Experience:** Investigate methods to improve the overall user experience, including user satisfaction, trust in the system, and ease of use. This may involve conducting user studies to gather feedback.

6. **Apply to Real-World Scenarios:** Demonstrate the applicability of the developed adaptive shared control systems in real-world scenarios, such as manipulation tasks, robotic assistive surgeries, and autonomous vehicles. This includes customizing the systems to address the specific challenges and requirements of different application domains.

By achieving these objectives, this research aims to contribute to the development of advanced human-robot collaboration systems that are not only technically robust but also user-friendly and adaptable to a wide range of tasks and environments.

1.4 Significance of the Study

The significance of this study lies in developing adaptive shared control systems, this research addresses several critical aspects that are pivotal for the future of human-robot interaction:

1. **Enhancing Task Performance and Efficiency:** Adaptive shared control systems can improve the efficiency and performance of tasks by leveraging the

strengths of both human cognitive abilities and robotic precision. This leads to more effective task execution in complex and dynamic environments.

2. **Improving Safety:** By dynamically adjusting control based on real-time data, these systems can enhance safety for both human operators and the environment in which the robots operate. This is particularly important in high-stakes scenarios such as surgical procedures, industrial automation, and disaster response.

3. **Boosting User Experience:** Developing intuitive interfaces and ensuring seamless interaction between humans and robots can greatly improve user experience and satisfaction. A positive user experience is crucial for the widespread adoption of robotic systems.

4. **Fostering Human-Robot Collaboration:** The research promotes a deeper understanding of how humans and robots can work together more effectively. This collaboration is essential for advancing fields such as assistive robotics, where robots support individuals with disabilities, and industrial robotics, where robots and humans share tasks in manufacturing processes.

5. **Advancing Technological Innovation:** The development of sophisticated control algorithms and learning from demonstration techniques contributes to the broader field of artificial intelligence and machine learning. These advancements can be applied to other areas of robotics and automation.

6. **Real-World Applicability:** The applicability of the developed systems to real-world scenarios showcases the practical relevance of the research. By demonstrating the effectiveness of these systems in real-world environments.

1.5 Limitations of the Research

While this research aims to make significant contributions to the field of adaptive human-robot shared control, it is important to recognize its limitations:

- **Generalizability:** The algorithms and systems developed in this research may be tailored to specific tasks or environments. While efforts are made to ensure versatility, the findings may not be universally applicable across all possible use cases in human-robot interaction.
- **Real-Time Processing:** The effectiveness of adaptive shared control systems relies heavily on real-time data processing. Limitations in computational power and latency may affect the performance of the systems, particularly in highly dynamic environments.
- **Human Factors:** Variability in human behavior and performance can introduce challenges in designing universally effective control systems. Differences in individual operator skills, preferences, and physiological responses need to be accounted for, which may limit the generalizability of the findings.
- **Interface Design:** User preferences and interface usability can vary widely. Designing interfaces that are intuitive remains a challenge, and user studies may reveal diverse opinions and needs.
- **Resource Constraints:** The research is conducted within the constraints of available resources, including time, funding, human subjects, and access to experimental equipment. These constraints may limit the scope of testing and the ability to explore all potential applications thoroughly.
- **Ethical and Safety Considerations:** Ensuring the safety and ethical use of adaptive shared control systems is paramount. The research may be limited

by the need to ensure safety, particularly when involving human subjects in experiments with real robots.

CHAPTER II

LITERATURE REVIEW

The field of adaptive human-robot shared control has seen considerable growth in recent years, with significant advancements in areas such as human-robot interaction, shared control, control allocation, learning from demonstration, and interface design.

This literature review aims to provide a comprehensive overview of the key research areas and recent developments that are relevant to this study.

2.1 Human-Robot Interaction

Human-robot interaction (HRI) is a multidisciplinary field focused on the development and evaluation of robotic systems that interact with humans. Recent research has highlighted the necessity of creating systems that are intuitive, safe, and efficient.

A survey study of HRI in [11] outlined the primary challenges and design principles. This work builds on earlier foundations, emphasizing how robots and humans can effectively interact in increasingly complex environments. Researchers in [12] further explored the concept of socially intelligent robots, focusing on recent advancements in robots that can understand and respond to human social cues. More recent studies, such as [13], have investigated the use of machine learning techniques to enhance interaction quality, demonstrating that robots can learn to adapt their behaviors based on human feedback.

The work in [14] discussed collaborative control, where the human and robot share control of a task, each contributing their strengths. This concept has evolved significantly, with recent works focusing on real-time adaptation of control strategies to improve collaboration.

2.2 Adaptive Robotics

Adaptive robotics has emerged as a key area of research, driven by the need for robots that can dynamically adjust to complex and changing environments.

Foundational theories and techniques for adaptive robotics were established through early research, which provided a thorough overview of adaptive control methods such as parameter estimation and adaptive filtering [3]. These contributions have been critical in understanding how robots can adapt to environmental and task-related changes.

A notable development in the field is the integration of learning algorithms that enhance robot performance based on experiential data. Frameworks have been introduced to enable robots to learn behaviors from demonstrations, utilizing deep reinforcement learning to adapt to complex scenarios [15]. These frameworks have shown considerable promise in allowing robots to generalize from observed tasks and adapt to new situations.

In industrial settings, the practical impact of adaptive robotics is evident. Adaptive control techniques have been emphasized for managing manufacturing processes and addressing unforeseen disturbances [6]. Integrating human expertise with adaptive algorithms has been highlighted as essential for maintaining operational efficiency in industrial environments.

Research has also demonstrated the enhancement of dynamic task execution in industrial robots through adaptive algorithms. These algorithms enable robots to adjust their actions in real-time based on environmental feedback, thereby improving performance in manufacturing tasks [16].

In the medical field, adaptive robotics have greatly improved surgical procedures. The importance of adaptive algorithms in upgrading robotic surgical systems has been highlighted, focusing on their role in boosting accuracy and flexibility within surgical settings [17].

2.3 Shared Control

Collaboration between humans and robots in shared control systems enables them to work together to accomplish tasks, with control distributed between them through arbitration or blending mechanisms [18].

Shared control in robotics, particularly in human-robot collaboration, presents distinct challenges that differ from those encountered in multi-robot systems. One primary difficulty lies in accommodating human unpredictability and creativity [19]. Unlike robots, which operate based on predefined algorithms and rules, humans bring an element of spontaneity and innovation to interactions, requiring adaptive strategies that can effectively integrate human input with robotic functions.

To address these challenges, various strategies have been developed to facilitate effective collaboration between humans and robots. These strategies often involve advanced sensor technologies and machine learning algorithms that enable robots to interpret and anticipate human actions, ensuring a seamless and intuitive interaction. For instance, gesture recognition systems allow robots to respond dynamically to human movements, while voice command interfaces enable natural language communication. Furthermore, adaptive control systems are designed to adjust the robot's behavior in real-time based on human feedback, enhancing the fluidity and efficiency of the collaboration. The ultimate goal is to create shared control systems that not only accommodate human unpredictability but also leverage human creativity to achieve complex tasks more effectively.

One simple strategy for shared control is direct control, where the human physically guides the robot through desired postures [20]. However, this approach is limited by its physical interaction and is not widely applicable in shared control systems.

Another approach is traded control, where control authority can be fully transferred to either the human or automated control algorithms, as seen in aircraft autopilot systems [21], for instance, during takeoff and landing phases, pilots might

take full control to handle the intricacies and immediate decision-making required, while during stable cruise flight, control is often transferred to autopilot systems to maintain altitude and direction efficiently. Traded control is not limited to aviation; it finds applications in other domains such as automotive systems, where adaptive cruise control allows drivers to alternate between manual driving and automated speed regulation. However, frequent switching between human and automated control can lead to inconsistencies in the system's behavior. Sudden changes in control authority might cause confusion or disorientation for the human operator, particularly if the transition is not smooth or well-coordinated.

Tactile control is another shared control strategy that transmits machine inputs to the operator through force feedback, allowing operators to respond to or resist these forces, as demonstrated in wheelchair control systems [22, 23]. Another example of tactile control can be seen in robotic surgery systems, where surgeons use haptic feedback to "feel" the resistance and texture of tissues while performing delicate operations.

The master-slave strategy, where the robot acts as the slave under human guidance, has regained popularity due to its potential to reduce human effort and enhance task performance, particularly in robotic-assisted surgeries [24, 25]. The master-slave configuration enables the surgeon to operate from a console, translating hand movements into precise manipulations of surgical instruments. This not only enhances the surgeon's dexterity but also provides a stable platform that reduces tremors and fatigue.

Policy blending approaches enable the integration of human and robot commands through task weight coefficients to adjust the influence of each source of input commands [26, 27].

Frameworks such as that proposed by [28] provide a structured approach to shared control, where agents aim to minimize individual cost functions representing error and

effort components, leading to a Nash equilibrium.

Overall, shared control methods aim to reduce workload and enhance operational efficiency by combining manual teleoperation with autonomous assistance, ensuring that human intervention remains essential [29].

2.4 Adaptive Shared Control

Adaptive shared control involves the dynamic adjustment of control authority between a human operator and a robotic system based on real-time data. This approach seeks to leverage the strengths of both human cognition and robotic precision.

Various shared control systems and their applications, such as teleoperation and assistive technologies, have been discussed in [30]. The necessity of understanding human intent and providing appropriate feedback has been emphasized.

The use of physiological signals, such as electroencephalography (EEG) and electromyography (EMG), to infer human intent and enhance the adaptability of shared control systems has been explored [31].

A Bayesian framework for shared control, which continuously updates the system's understanding of the human's goals based on their actions, has been proposed [32]. This method has been shown to improve task performance and reduce cognitive load on the operator.

The integration of augmented reality (AR) into shared control systems has been discussed [33, 34]. AR provides intuitive visual feedback, which enhances the operator's situational awareness and reduces cognitive load.

2.4.1 Applications of Adaptive Shared Control

Adaptive shared control has been applied in various domains, including industrial robotics, assistive technologies, and autonomous vehicles.

[21] discussed the role of adaptive shared control in industrial robotics, particularly in improving efficiency and safety in collaborative manufacturing workcells. They

emphasized the potential for reducing physical and cognitive demands on human workers through adaptive control strategies.

In assistive technologies, [35] developed an adaptive shared control system for wheelchair navigation, combining user input with autonomous planning to improve mobility for individuals with disabilities.

In the domain of autonomous vehicles, [36] explored the use of shared control to improve the safety and comfort of self-driving cars. Their system dynamically adjusts the level of autonomy based on the driver's state and environmental conditions, demonstrating significant improvements in overall driving performance.

2.5 Human Intention Estimation

Estimation of human intention is a key element in shared control scenarios, crucial for enabling robots to effectively collaborate with humans. In shared control frameworks, learning human intention is challenging due to the inherently complex and sometimes unpredictable nature of human behavior.

Human actions are often goal-directed [37], and effective collaboration requires both humans and robots to work toward the same goal. However, predicting human intention precisely under all circumstances is difficult as different operators may use varying strategies to accomplish a task, and even the same operator might change approaches with repetition. Furthermore, the level of human expertise further complicates intention estimation, as practice alters movement patterns [38].

[39] provided a comprehensive review of intention estimation techniques, categorizing them into explicit and implicit methods. Explicit methods involve direct communication from the user, such as through verbal commands or manual input. Implicit methods infer intention based on observable behavior and physiological signals. The most straightforward way for human intention estimation is to explicitly

provide the human goal to the robot [40]. However, in dynamic tasks requiring continuous adjustments, this approach may not be practical [41, 42, 43, 44, 45].

Recent advances have explored various strategies to improve intention estimation. [46] highlighted the use of machine learning to predict user intent in real time by processing sensor data, such as from cameras and motion tracking devices, enhancing the responsiveness and accuracy of shared control systems. Similarly, [47] demonstrated the effectiveness of neural networks in processing complex sensor data to predict human actions, resulting in significant improvements in accuracy and robustness.

Visual information is commonly used to estimate human intention, often obtained through cameras or vision sensors [48, 44]. However, human behavior context-dependence and ambiguity can make visual cues alone insufficient for accurate intention estimation, and real-time processing can be computationally intensive, leading to delays and misinterpretations [49].

Alternatively, wearable sensing devices equipped with sensors capture physiological and kinematic data, helping to infer intentions [50]. However, scalability to larger systems or more complex tasks remains a limitation with wearable sensing devices [51, 52].

To address the importance of context in intention estimation, [53] emphasized understanding the situational context, proposing a framework that combined environmental data with user behavior for better accuracy. [54] introduced multimodal data fusion, merging visual, auditory, and physiological sensor data to achieve greater reliability in predicting user intent.

When direct communication or wearable devices are not feasible, robots can estimate human intentions during task execution by collecting sufficient data or continuously observing the task state until completion [55, 56, 57].

Using these human intention estimation methods, shared control systems can better exploit both human and robotic strengths, creating a synergistic partnership that

improves overall control performance, robustness, and accuracy while reducing the workload of human operators [58, 59].

2.6 Learning from Demonstration

Learning from demonstration (LfD) is a technique where robots learn tasks by observing demonstrations. The choice of LfD over other robot learning methods is compelling when optimal behavior can be neither easily scripted nor easily defined as an optimization problem but can be demonstrated. Demonstration approaches mostly fall into three categories, kinesthetic teaching, passive observation, and teleoperation. [60] provided a comprehensive survey of LfD techniques, categorizing them based on demonstration types and learning algorithms.

Kinesthetic teaching enables the user to demonstrate the task by physically moving the robot through the desired movements [61, 62]. Kinesthetic teaching depends on the development and maintenance of the robot hardware and does not need additional sensors and interfaces, which simplifies the machine-learning process. There are several limitations with this approach; main limitation of kinesthetic teaching is that the expert user must often use more of their own degrees of freedom to move the robot than the number of degrees of freedom they are trying to control. In addition, the quality of the demonstration depends on the dexterity and smoothness of the expert demonstrator.

Passive observation is where the robot learns from passive observation of the user [63]. In this approach, the user performs the task using their own body, while the robot observes and tracks the motions of the user's body, which could be pinned with markers to facilitate tracking. Passive observation is suitable for application to robots with many degrees of freedom, where kinesthetic teaching is difficult or not feasible, however, the machine learning part of the problem usually becomes more complicated as it is required to encode or learn a mapping of human user actions

to those executable by the robot. Sensor noise in the observations is also one of the important challenges in demonstration, but learning from passive observation has been successfully applied to different tasks such as autonomous driving [64] and knot tying [65].

Teleoperation is another way of demonstration that requires an external input to the robot through an interface such as a mouse, joystick, pedal, graphical user interface, or haptic devices. This has been widely used for task and trajectory learning. The limitations of teleoperation include the additional effort required to develop the interface input and teach the user to perform the task via the interface [38]. One of the advantages of teleoperation is that it can be easily coupled with simulation for data collection and experimentation. Another significant benefit is that it enables operators to demonstrate tasks remotely.

In this area, recent advances include the integration of reinforcement learning with LfD to improve adaptability and robustness. [66] introduced a framework that combines these techniques to enable robots to adapt to new environments and tasks.

[29] explored the use of deep learning in LfD, demonstrating that deep neural networks can effectively capture the nuances of human demonstrations and improve the robot's ability to replicate complex behaviors.

2.7 Interface Design for Human-Robot Systems

The design of user interfaces is crucial for an effective human-robot interaction. Interfaces must be intuitive, provide the necessary feedback and not overwhelm the user.

[67] discussed principles of interface design for shared control, emphasizing transparency and predictability. They argued that the system should clearly communicate its actions and intentions to the human operator to build trust and improve collaboration.

[68] explored the use of AR interfaces, demonstrating their effectiveness in providing intuitive visual feedback and improving operator situational awareness.

Recent studies by [69] and [70] have focused on user-centered design approaches, involving end users in the design process to ensure the interfaces meet their needs. These studies highlight the importance of iterative design and testing to create effective interfaces.

2.8 Human to Robot Skill Transfer

Learning from demonstration has been applied on various robotic tasks where the robot can either mimic the motion of the demonstrator or learn how the demonstrator acts, reacts, and handles errors [71]. LfD can implicitly learn the task requirements and constraints from the demonstrator that can enable adaptive behavior. In LfD, trajectories can be learned by probabilistic methods such as dynamic movement primitives (DMPs), probabilistic movement primitives (ProMPs), Gaussian mixture regression (GMR), Gaussian process regression (GPR) and neural processes (NPs).

Dynamic movement primitives (DMPs)[72] are a framework for learning a trajectory from a demonstration. DMPs encode the demonstrated trajectory as a set of differential equations. The trajectory can be efficiently learned after only one demonstration and can be adapted to new goal positions and time duration. However, DMPs do not handle variation and uncertainty information and generalization to new, unseen situations in DMPs is limited. They have been used in different robotic scenarios, such as the reproduction of the human walk for exoskeletons [73], collaborative bimanual tasks [74].

Probabilistic movement primitives (ProMPs) [75] can also learn the linear relationship between environment variables and complex trajectories. ProMp encodes a distribution of trajectories and generates stochastic policies, in contrast to DMP

approaches that can only represent the mean solution, which is known to be sub-optimal. Both DMPs and ProMPs encode trajectories by relying on basis functions equally spaced in time; selecting appropriate basis functions becomes difficult with high-dimensional inputs.

Gaussian mixture regression (GMR) [76] uses the Gaussian conditioning theorem to estimate the distribution of the output data given input data, GMR is trained with an Expectation Maximization algorithm. A Gaussian mixture model (GMM) [77] is first estimated to encode the joint distribution of input and output data points. Given an observed input, the output is then predicted via a linear combination of conditional expectations. In medical robotics, GMM is used in [78, 79].

Gaussian process regression (GPR) [80] is a nonparametric Bayesian approach. Infers a probability distribution over all possible values. Gaussian processes regression, instead of calculating the probability distribution of parameters of a specific function, calculates the probability distribution over all possible functions that fit the data. Unfortunately, Gaussian processes become computationally intractable as the size or dimension of the dataset increases [81].

Neural Processes (NPs) [82] approach regression by learning to map a context set of observed input-output pairs to a distribution over regression functions.

Conditional Neural Processes (CNPs) [1] have the benefit of fitting observations efficiently with linear complexity in the number of context input-output pairs and can learn predictive distributions conditioned on context sets. However, a fundamental drawback of CNPs is that they tend to underfit, leading to inaccurate predictions at the input of the observed data on which they are conditioned.

In Attentive Neural Processes (ANPs) [83] this problem is addressed and an attention mechanism is added to the neural processes to improve the accuracy of the predictions and also expand the range of functions that can be modeled.

Conditional neural movement primitives (CNMPs) [2] is another framework built

on top of CNPs [1], which is designed as a robotic movement learning and generation system. CNMPs can be conditioned on single or multiple time-steps to produce trajectory distributions to satisfy the given conditions.

In CNPs and CNMPs Having a set of inputs (observations) $O(x_i, y_i), i = 1 - n$ where n is a chosen parameter indicating the number of observation tuples, and a set of points to estimate their y values (targets) Tx_i we would like to learn a conditional distribution $Q(f(T)|O, T)$ over all the functions $f : X \to Y$.

The CNP architecture has three main components, decoder, aggregator and Encoder, which are shown in Figure 1.

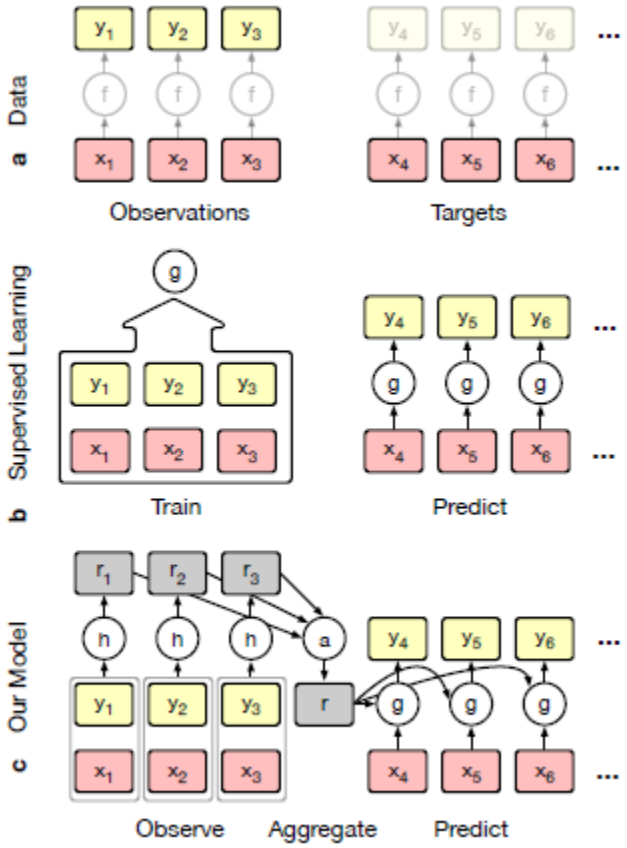


Figure 1: Conditional neural processes architecture. Image credit: [1]

Encoder, which is a multi layer perceptron (MLP) with ReLU activation function,

takes some (x, y) pairs as context points, in and results in an individual representation for each pair (equation (1)).

$$r_i = E(x_i, y_i) \quad \forall (x_i, y_i) \in O \tag{1}$$

Aggregator combines all the individual representations and results in a general representation which contains the information about the underlying unknown function that maps the inputs to outputs (equation (2)).

$$r = r_1 \oplus r_2 \oplus ... \oplus r_n \tag{2}$$

Decoder which is also an MLP with ReLU activation function takes the general representation and some x_t values as targets, and results in the mean and variance of the estimated output (equation (3)).

$$(\mu_t, \sigma_t^2) = D(x_t, r) \quad \forall (x_t) \in T \tag{3}$$

The time complexity at estimation time is only $O(n + m)$, for making m predictions (for target points) with n observations, which is much faster compared to other stochastic methods such as Gaussian Processes.

The CNMPs architecture is shown in Figure 2.

One challenge faced by Conditional Neural Movement Primitives (CNMPs) is their difficulty with extrapolation, where the accuracy of predictions significantly decreases when the model encounters data outside the range it has been trained on. This limitation arises because CNMPs primarily focus on interpolating between known context points, making them less reliable when asked to generalize to unseen regions. To address this, Adaptive Conditional Neural Movement Primitives (ACNMPs) [84] extend CNMPs by incorporating reinforcement learning to overcome the training set range limit. In case of extrapolation or failure, a reinforcement learning (RL) algorithm is

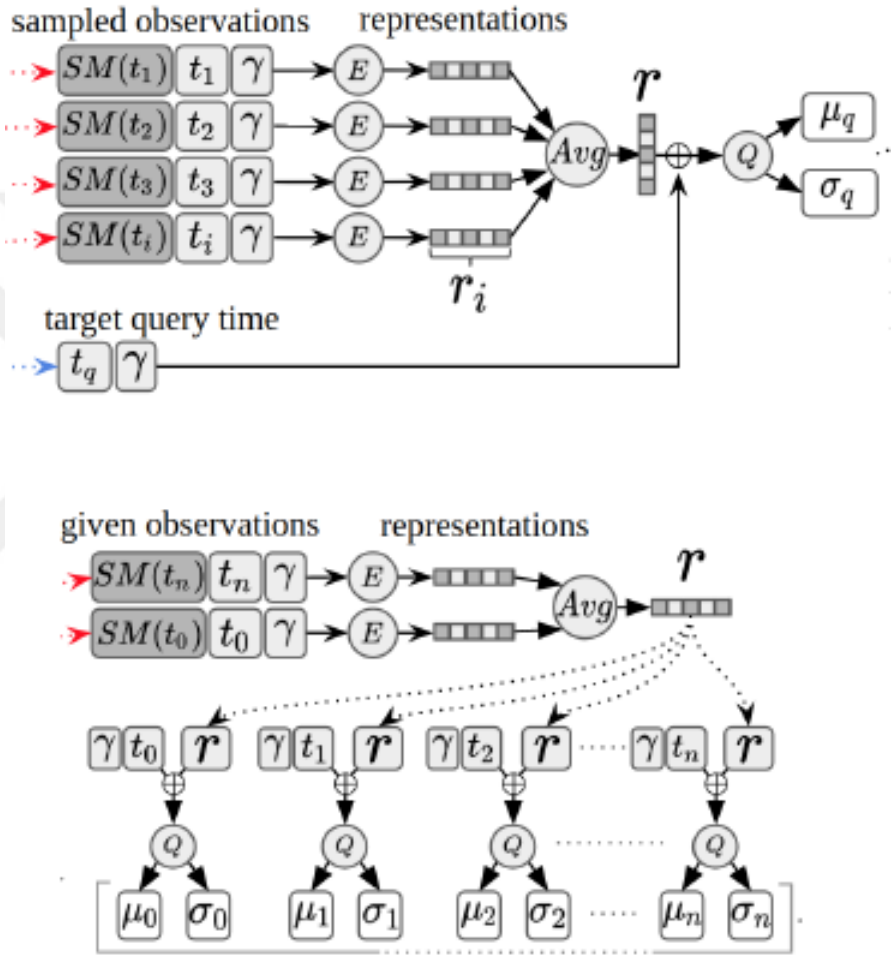


Figure 2: Conditional neural movement primitives. Image credit: [2]

used to find a trajectory that satisfies the conditions while preserving the characteristics of the observations. The new generated trajectory by RL will be then added to the training set to increase generalization.

2.9 Challenges in Adaptive Human-Robot Shared Control

Adaptive human-robot shared control systems aim to seamlessly integrate human input with autonomous robotic actions, adapting to dynamic environments and user needs.

[85] highlighted the need for comprehensive testing and validation methods to ensure the safety and efficacy of adaptive shared control systems. They also emphasized ethical considerations, such as privacy and accountability, which are critical for the widespread adoption of these technologies.

However, several challenges must be addressed to achieve effective and reliable shared control:

1. **Understanding Human Intent:** Accurately interpreting human intentions in real time is complex. This requires sophisticated algorithms that can process various signals, such as haptic commands, gestures, movement trajectories, voice commands, or physiological data, to determine the user's goals and preferences.

2. **Balancing Autonomy and Control:** Deciding the level of control to allocate between the human and the robot is crucial. Too much autonomy can make users feel out of control, while too little can negate the benefits of automation. Finding the right balance that adapts to different tasks and situations is a significant challenge.

3. **Robust Communication:** Ensuring effective communication between humans and robots is essential for shared control. This involves developing intuitive

interfaces and feedback mechanisms that allow users to understand the robot's actions and intentions clearly.

4. **Real-time Adaptation:** Adaptive systems must respond quickly to changes in the environment and user input. This requires efficient computational methods and algorithms capable of real-time processing to adjust control strategies on the fly.

5. **Safety and Reliability:** Ensuring safety in shared control systems is paramount, especially in scenarios where humans and robots work closely together. Systems must be reliable with fail-safes to handle unexpected situations or errors.

6. **User Training and Acceptance:** Users may need training to effectively interact with shared control systems. Ensuring that these systems are user-friendly and gaining user trust and acceptance is an important matter.

7. **Handling Uncertainty:** Adaptive systems must be able to deal with uncertainties in human behavior and the environment. This involves developing models that can predict and adapt to various scenarios and user actions.

8. **Integration of Multimodal Inputs:** Shared control systems often rely on multiple input modalities, such as vision, touch, and sound. Integrating these inputs in a coherent manner to form a comprehensive understanding of the user's intent is challenging.

CHAPTER III

HUMAN ADAPTATION TO HUMAN-ROBOT SYSTEMS

The study presented in this chapter builds upon foundational work that was partially published in my master's thesis and further elaborated in [38]. The initial framework and findings from these previous publications [86, 38] provided the essential groundwork that enabled the advanced research and analysis found in the subsequent chapters of this PhD thesis. This progression of research demonstrates the evolution and deepening of the concepts and methodologies, ultimately contributing to the advancements detailed in this dissertation.

Human adaptation to shared control involves several cognitive and motor processes. Initially, the human operator must understand and predict the robot's behavior and integrate this understanding with their own actions. This requires the operator to adjust their control strategies based on feedback from the robot, leading to a dynamic interplay between human and robot.

Several factors influence the effectiveness of human adaptation in shared control systems. These include transparency of the robot's goal and actions, intuitiveness of the control interface, and the adaptability of the control sharing mechanisms. Transparent communication of the robot's actions can significantly reduce the cognitive load on the human operator, allowing for more fluid and effective collaboration. In addition, intuitive interfaces that closely align with human motor control can facilitate quicker adaptation and reduce errors.

The human adaptation process can be examined through various lenses, including learning curves, task performance metrics, and subjective user experience. Research has shown that with practice, humans can significantly improve their performance in

shared control tasks, achieving a high level of synergy with the robotic system[38]. However, the rate and extent of this adaptation can vary widely between individuals, influenced by factors such as prior experience with similar systems, individual cognitive and motor skills, and the specific design of the shared control system.

Understanding human adaptation to shared control is essential for designing systems that are not only efficient but also user-friendly and accessible to a broader range of users. By studying how humans learn and adapt to these systems, we can develop more effective training protocols, optimize control algorithms, and create interfaces that facilitate smoother and more intuitive human-robot collaboration.

In this chapter, we explore the dynamics of human adaptation to shared control through a series of experiments involving a ball-balancing task. We analyze how the shared control mechanism impacts task performance and human learning. Furthermore, we investigate the subjective experiences of participants as they adapt to the shared control system. The findings provide insights into the key factors that influence successful human adaptation to human-robot shared control systems.

3.1 Shared Control Framework

The framework used in this chapter enables the robot to assist a human in completing a task by predicting the human's intentions. As the human initiates the task, the robot concurrently estimates the human's goal and contributes to the control process by generating commands that help the human towards achieving the predicted intention. The combined control command that drives the robot is a mix of human and robot inputs. In this approach, we use a convex combination of these inputs to determine the final command sent to the robot. Specifically, the net command is calculated as $C_{net} = \omega C_H + (1 - \omega)C_R$, where ω represents the weight parameter to share the control command, C_H is the human command and C_R is the robot command. Alternative control sharing methods, such as adaptive control sharing, are

also feasible.

3.2 Designed Task

To assess how the human adapts to the shared control framework, a ball balancing task was designed. In this task, a robotic arm holds a tray and a ball is placed in the center of the tray at the start. The position of a target is marked on the tray, and the objective is to move and balance the ball on the target by tilting the tray, controlled by a computer mouse; while the control command is a convex combination of human and robot commands with equal weights (ω is set to 0.5 in $C_{net} = \omega C_H + (1-\omega)C_R$)).

3.3 Robotic Setup

A 6DOF Kuka Agilus R6000 robotic arm was used to hold the tray, which measures 70cm by 70cm. The tray was attached to the robotic arm's end effector, with two joints functioning as wrist and elbow joints to tilt the tray along two axes. A ball with a 3cm radius was placed on the tray, and its position and velocity were tracked in real-time by an OptiTrack infrared camera system operating at 50 Hz.

The camera first detects the edges and center of the tray and then scales the detected positions to the actual dimensions of the tray to determine the position of the ball.

3.3.1 Robot Teleoperation

To allow human control of the robotic arm to tilt the tray, a standard computer mouse is used. The horizontal and vertical movements of the mouse are linearly mapped to the desired angular movements of the wrist and elbow joints, respectively, tilting the tray to move the ball accordingly. The control operates at 25Hz, and mouse movements are experimentally scaled to provide intuitive control.

$$\theta_{\text{wrist-desired}} = \theta_{\text{wrist-current}} + k\Delta x \tag{4}$$

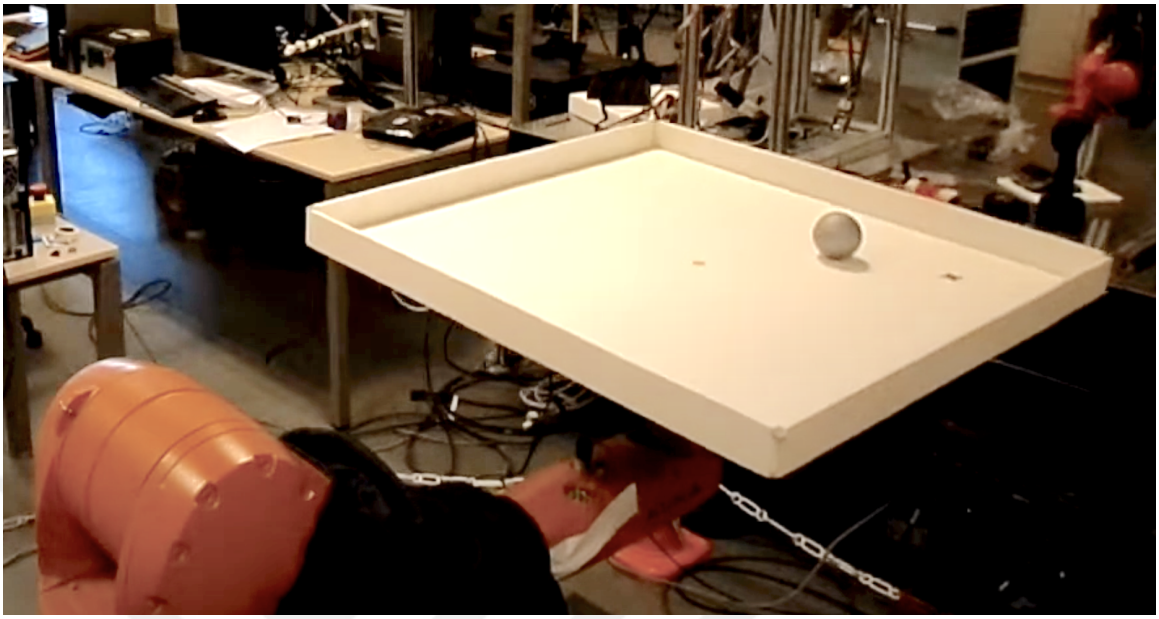


Figure 3: Ball balancing task robotic setup

$$\theta_{\text{elbow-desired}} = \theta_{\text{elbow-current}} + k\Delta y \tag{5}$$

Here, θ_{wrist} and θ_{elbow} represent the joint angles, k (set to 1) is the scaling constant, and Δx and Δy are the mouse displacements.

3.3.2 Autonomous Controller

The robot generates commands using an autonomous controller designed to move and balance the ball in the predicted target position. This controller is developed through learning from demonstration. The task state is defined as:

$$S = [x, y, V_x, V_y, J_{wrist}, J_{elbow}, \omega_{wrist}, \omega_{elbow}] \tag{6}$$

Here, (x, y) and (V_x, V_y) denote the ball's position and velocity, respectively, while (J_{wrist}, J_{elbow}) are the robot's joint angles, and $(\omega_{wrist}, \omega_{elbow})$ are the joint's angular velocities.

Assuming a linear relationship between control commands and the state, the control policy can be approximated using linear regression. To collect demonstrations, an expert performed the task multiple times, while the state of the task and the corresponding commands were recorded. The recorded data formed matrices S (state) and C (command). The weight matrix W mapping the states to commands is found using the pseudo inverse of S:

$$W = S^{+}C \tag{7}$$

With W known, the autonomous controller's command c can be calculated from the current state s:

$$c = sW \tag{8}$$

Figure 4 shows the predicted and actual commands of the expert. Data from 200 seconds of expert demonstration were used to develop a controller capable of balancing the ball at any tray location by mapping the desired target point to the tray's center.

3.3.2.1 Human Intention Prediction

In this experiment, human intention refers to the target position in which the human aims to move and balance the ball. The prediction begins after collecting 1000 data points (4 seconds) from the task starting time. The ball position within this window is modeled as a Gaussian distribution:

$$P \begin{bmatrix} x \\ y \end{bmatrix} \sim N(\mu, \Sigma) \tag{9}$$

where (x, y) represents the ball position. The mean matrix μ is defined as:

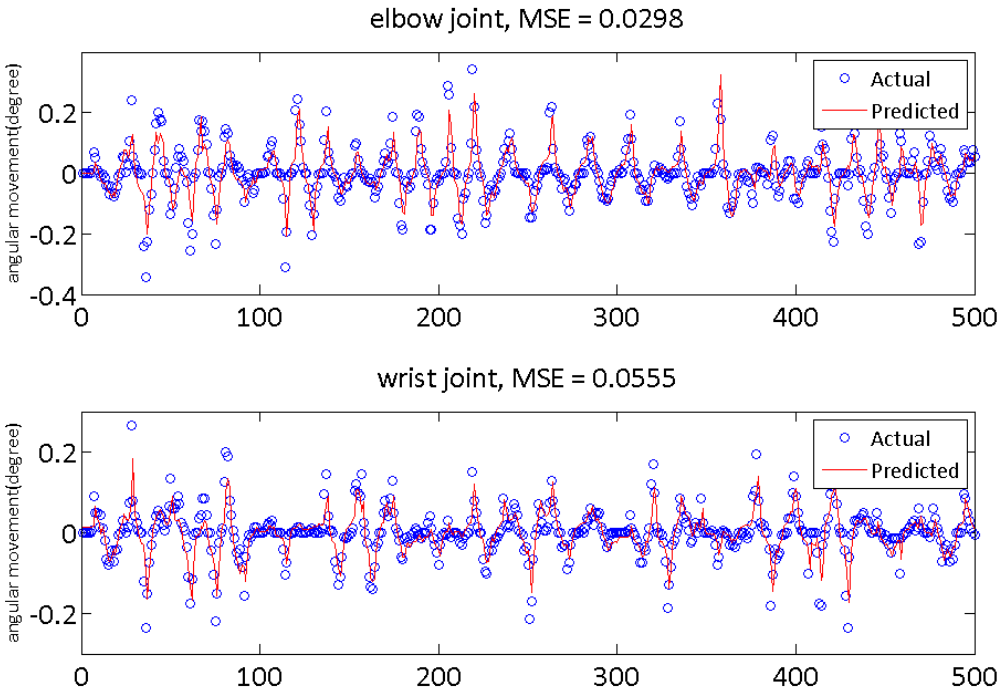


Figure 4: Actual and predicted commands obtained through linear regression for the robot's wrist and elbow. For clarity, only 500 data points (collected over 2 seconds) are shown, with MSE = 0.0298 for the wrist and MSE = 0.0555 for the elbow joint.

$$\mu = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} \tag{10}$$

with μ_x and μ_y being the mean positions of the balls along the x and y axes. The covariance matrix Σ is:

$$\Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix} \tag{11}$$

where $\Sigma_{xx} = \text{var}(x)$, $\Sigma_{yy} = \text{var}(y)$, and $\Sigma_{xy} = \Sigma_{yx}^T = \text{cov}(x, y)$.

The Gaussian mean represents the estimated goal position, and the variance indicates the confidence of prediction.

3.3.3 Shared Controller

In the shared control condition, both humans and robots generate control commands. The human initiates the task, while the robot assists by inferring the human's intention (modeled as a continuous two-dimensional target position and augmenting the human's commands. The final command driving the robot is a convex combination of human and robot inputs.

3.4 Experiments

To evaluate the impact of shared control on task performance and human learning, experiments were conducted under two conditions: human control and shared control.

Human Control Condition: In this condition, only the human's teleoperation commands control the robot.

Shared Control Condition: Here, both human and robot contribute to generating the control commands. The human starts the task, while the robot assists by inferring the human's goal (modeled as a continuous two-dimensional target position)

and augmenting the human's commands. The final command driving the robot is a convex combination of the human and robot inputs.

3.4.1 Participants

Twenty naive subjects (12 men, 8 women) participated in the experiment. They were students from the engineering and psychology faculties at Özyeğin University, Turkey, matched by age and gender and divided into two groups for the human control and shared control conditions. The human control group included six men and four women with an average age of 25.8 (range: 22-28 years). The shared control group included six men and four women with an average age of 25.9 (range: 23-31 years).

3.4.2 Experiment Design

Four target positions were marked on the tray as potential targets for the subjects. However, the robot was not provided with information about these target positions. At the start of each experimental trial, the ball was placed in the center of the tray with zero velocity. The subjects were tasked with moving and balancing the ball in one of the four marked target positions. Each experimental session consisted of four sub-sessions (blocks), with each sub-session comprising four experimental trials, totaling 16 trials per session. To avoid fatigue, subjects could take short breaks between sub-sessions. Each subject participated in four experimental sessions on separate but consecutive days, resulting in 64 trials per subject. This design aimed to monitor their learning progress.

At the beginning of the first session, the subjects received instructions detailing the task, how to use the interface, the start time, and when the task was finished. Subjects were not informed whether they were performing the task under human control or under shared control conditions. Those in the shared control group were unaware of the robot's involvement. At the end of the final session, the subjects completed a demographic information questionnaire.

3.4.3 Performance Measure

The performance of individual subjects in the ball balancing task was measured using the length of the trajectory. The length of the trajectory refers to the length of the path the ball travels on the tray in one trial, starting from the center of the tray to the desired target point. A shorter trajectory length indicates higher task performance.

3.4.4 Analysis Method

To compare the performance of subjects under the two conditions (Shared Control vs. Human Control) based on their trajectory length, we used one-way MANOVAs for each block within a day and for daily performance over four days. For these analyses, we summed the four trials for each block and then the four blocks for each day. Further analysis was performed to inspect the learning progress of the subjects under the two conditions.

3.5 Results

3.5.1 Performance Results

In this section, we report and compare the performance of subjects under human control and shared control conditions.

To determine whether there were any group mean differences in performance between the shared control and human control conditions, we performed a one-way MANOVA for four days' performance. As shown in Table 1, there was no significant difference in trajectory length performances between the two conditions on the first and second days. However, a significant difference in the mean of the group was observed on the third and fourth days, the shared control group showing lower trajectory lengths than the human control group.

To investigate whether performance was the same in shared control and human control on the first day, we conducted a one-way MANOVA. Although MANOVA was

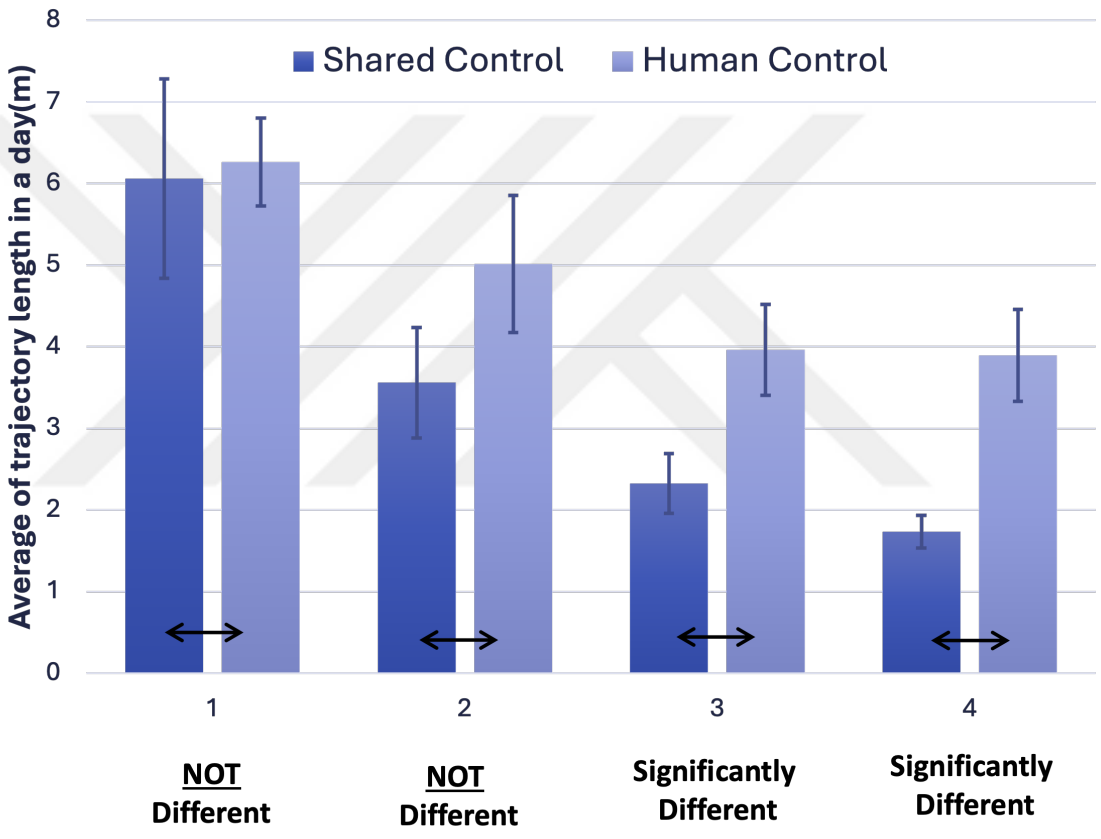


Figure 5: Means and Standard Error of the Mean for day1, day2, day3, and day4 performance of human control and shared control conditions (Error bars: SEM)

Table 1: Statistical analysis results of shared and human control conditions over different days.

Days	Condition	Mean(mm)	Stdv(mm)	ANOVA
day1	Shared	6138.8013	4003.3009	$F(1,18) = 0.005, p = .943, \eta^2 = .000$
	Human	6238.4168	1701.2981	
day2	Shared	3450.5946	2028.8708	$F(1,18) = 2.171, p = .158, \eta^2 = .108$
	Human	5005.7661	2650.0819	
day3	**Shared**	**2298.1561**	**1057.6340**	$F(1,18) = 6.248, p = .022, \eta^2 = .258$
	Human	**3908.4740**	**1741.2778**	
day4	**Shared**	**1733.2929**	**623.7290**	$F(1,18) = 13.193, p = .002, \eta^2 = .423$
	Human	**3894.1631**	**1742.9222**	
Wilks' Lambda = .500, $F(4,15) = 3.75$, $p = .026$, $\eta^2 = .50$				

significant, the follow-up ANOVAs for each block were not significant. This indicates that there are no significant differences between the shared control and human control groups in trajectory length performance for any of the four blocks on the first day.

To assess whether there were differences in trajectory length performance across the four days of the experiment for each condition, we performed several paired-samples t tests for shared control and human control conditions separately. Under the shared control condition, the trajectory length performance improved significantly every day from day 1 to day 4.

In human control conditions, trajectory length performance improved significantly from day 1 to day 2, day 1 to day 3, and day 1 to day 4 and day 2 to day 3, and day 2 to day 4, but not from day 3 to day 4.

3.5.1.1 Interpretation

Task performance in the human control condition plateaued between days 3 and 4, suggesting that learning stopped on day 3. In contrast, the shared control group showed significant progress from day 3 to day 4.

3.5.2 Learning Progress Results

In this section, we explore the learning progress of individual subjects and compare their adaptation rates between human control and shared control conditions to understand how quickly they learn the task and adapt to the situation.

We tracked subjects' learning progress by fitting linear, exponential, and power functions over the performance data for all 64 trials for each subject and reported the R-Squared value.

Table 2: R-Squared Value of the fitted regression functions comparison of Linear, Exponential, and Power fit for Shared Control and Human Control conditions.

Condition	Subject	Linear	Exp	Power	Best fit
Shared Control	1	0.33	0.5478	0.4861	Power
	2	0.2762	0.3833	0.5003	Power
	3	0.2104	0.1981	0.2547	Power
	4	0.1393	0.1116	0.1761	Power
	5	0.388	0.407	0.3014	Exp
	6	0.2588	0.2777	0.2779	Power
	7	0.1623	0.0767	0.0732	Exp
	8	0.1498	0.0794	0.1492	Linear
	9	0.2153	0.2333	0.3057	Power
	10	0.2196	0.2254	0.2723	Power
Human Control	1	0.016	0.0069	0.0527	Power
	2	0.1343	0.1593	0.1908	Power
	3	0.0939	0.0391	0.1424	Power
	4	0.2052	0.1065	0.1101	Linear
	5	0.1267	0.0095	0.0946	Linear
	6	0.002	0.0008	0.0026	Power
	7	0.133	0.1747	0.1606	Linear
	8	0.0043	0.047	0.1606	Power
	9	0.1885	0.1656	0.2259	Power
	10	0.0495	0.0172	0.0238	Linear
Average		0.16518	0.165094	0.19021	Power

The power function yielded the highest R-Squared values for all subjects in both control conditions, indicating that it best explained the learning trend. The fitted power functions for all subjects are shown in Table 2.

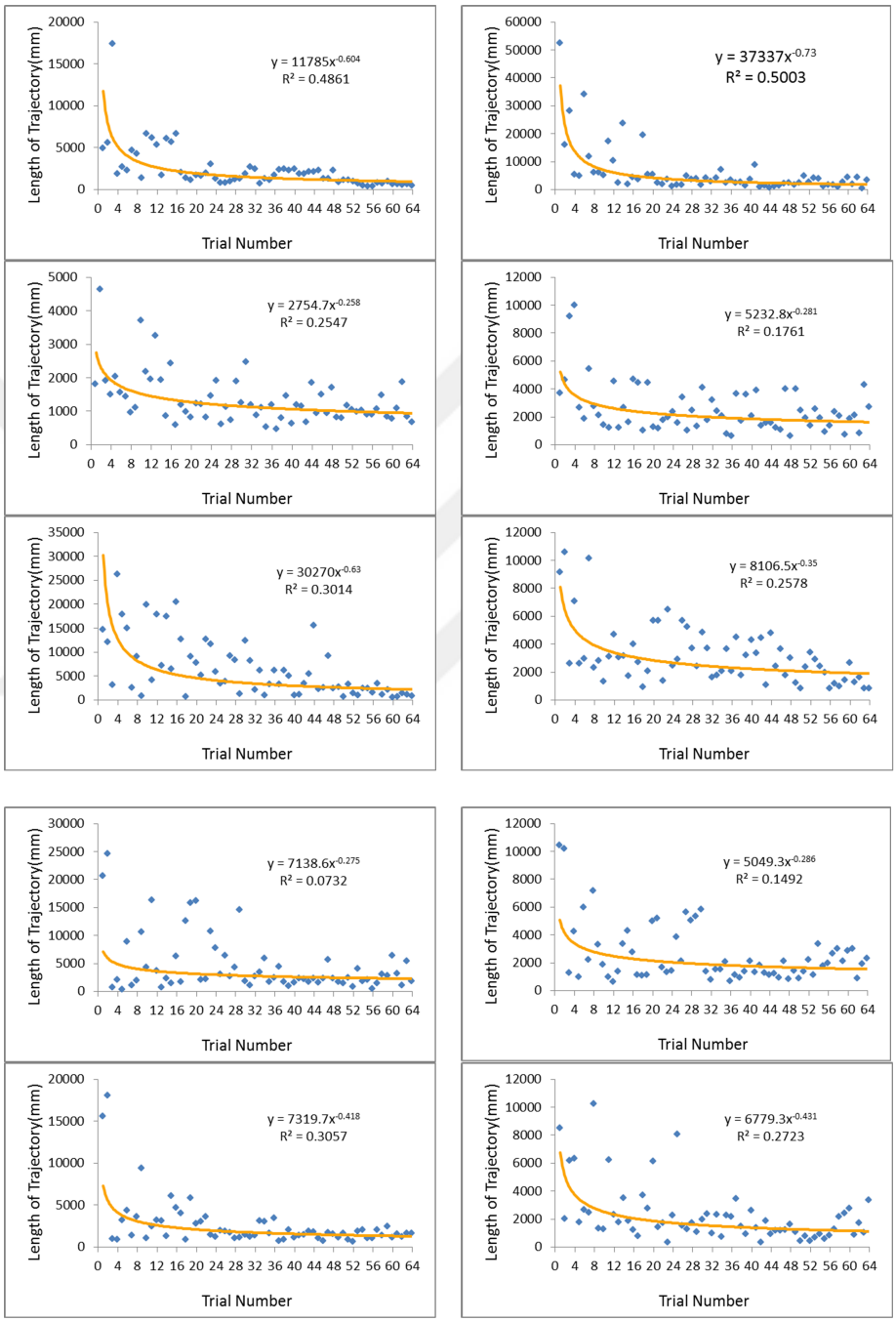


Figure 6: Fitted power function on the performance of shared control group

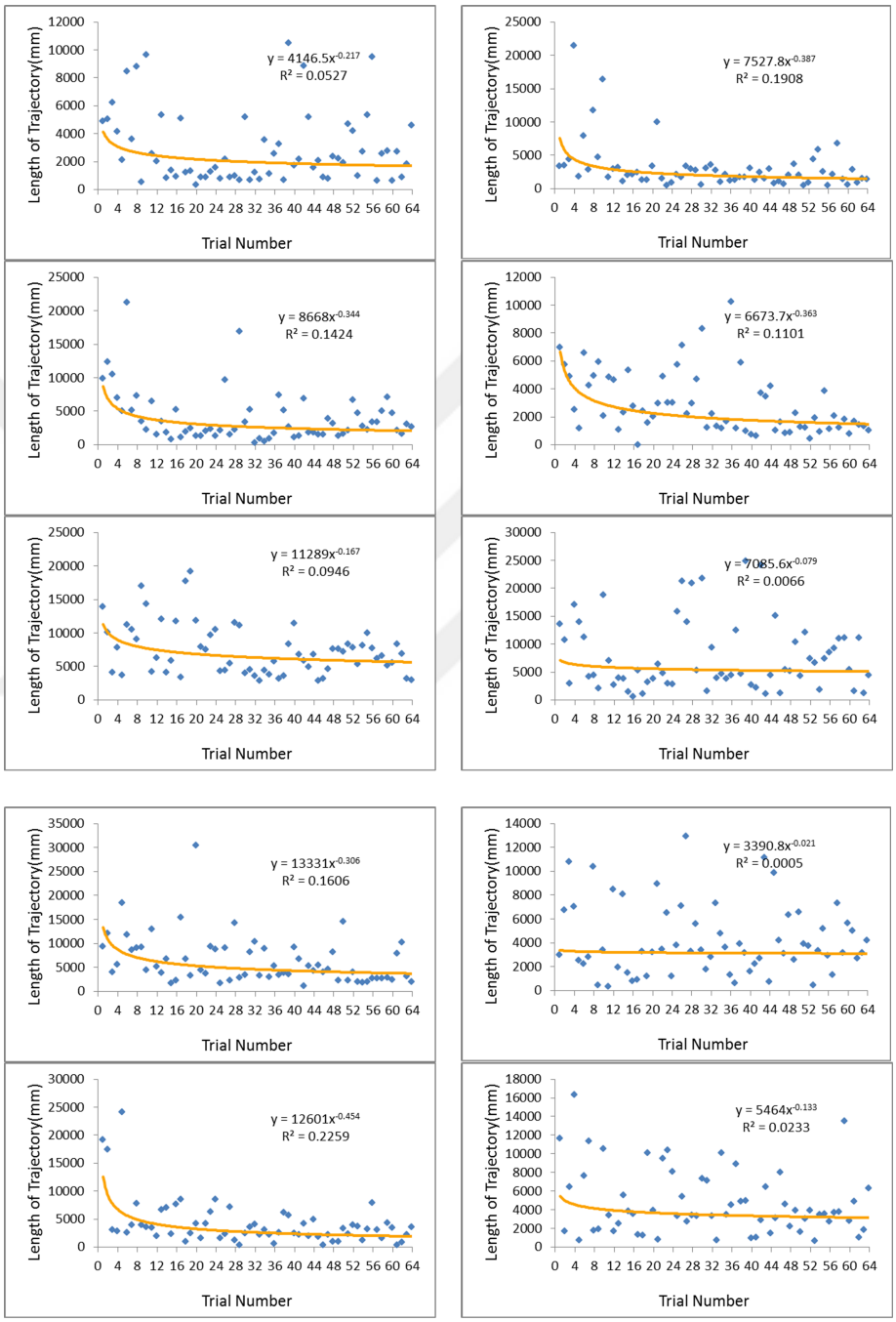


Figure 7: Fitted power function on the performance of human control group

3.5.2.1 Learning Rate

We defined the additive inverse of the power parameter as the learning rate for each subject. The learning rate values are reported in Table3.

Table 3: Learning rates of subjects under different control conditions

Shared Control		Human Control	
Subject	**Learning rate**	**Subject**	**Learning rate**
Subject 1	0.4343	Subject 11	0.2548
Subject 2	0.3498	Subject 12	0.3756
Subject 3	0.4025	Subject 13	0.1207
Subject 4	0.3956	Subject 14	0.2653
Subject 5	0.4548	Subject 15	0.2865
Subject 6	0.3649	Subject 16	0.3971
Subject 7	0.3189	Subject 17	0.0472
Subject 8	0.4902	Subject 18	0.3465
Subject 9	0.5574	Subject 19	0.1347
Subject 10	0.5342	Subject 20	0.2548
Average	**0.4263**	**Average**	**0.2471**

Subjects in the shared control condition exhibited a higher average learning rate compared to those in the human control condition as reported in Figure 8.

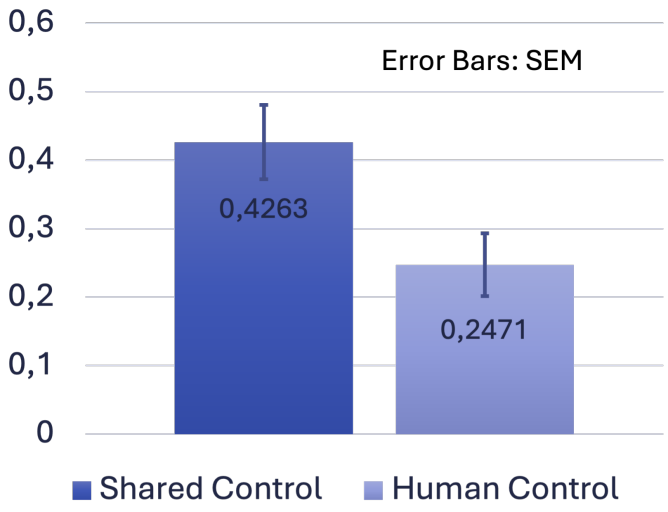


Figure 8: Mean of Learning Rate for shared control condition and human control condition, and Standard Error of the Mean.

We conducted a T-test to compare the learning rates between the shared control and human control conditions in Table. 4.

Table 4: Statistical comparison between Shared Control and Human Control groups.

	Shared Control	Human Control	diff	95% Confidence Interval
Mean	0.4263	0.2471	0.179	(0.030, 0.329)
Variance	0.02948	0.020932		
Observations	10	10		
Hypothesized				
Mean Difference		0		
df		17		
t Stat		2.530		
P(T¡=t) two-tail		0.022		
T Critical two-tail		2.110		
Reject Null Hypothesis because $p < 0.05$ (Means are Different)				

The results of the T-test indicated a significant difference in the learning rates between the two conditions.

3.5.3 Survey

At the end of the final experimental session, the subjects completed a survey. The responses to Question 10 were categorized into three classes: negative (blocking), neutral (no help), and positive (helping) to assess their perception of the robot's role in the task. In the shared control group, 9 out of 10 subjects noticed the robot's participation, with 8 finding it helpful and 1 finding it obstructive when attempting rapid movements. In the human control condition, 9 out of 10 subjects reported a neutral experience, as expected.

In the shared control condition, some subjects initially exhibited high exploration, which allowed them to better observe and learn the robot strategy and react accordingly, leading to better collaboration and higher task performance. The performance of one such subject is illustrated in

Analysis of data from the shared control group revealed that skilled performers

could guide the robot toward their objectives after comprehending its behavior. A survey response from a participant emphasized this learning curve:

> **Q9:** Do you think you have become an expert in this task?
>
> **A:** Maybe I can say that with further practice I will become an expert with the help of the robot. The more I could learn its behavior and my own shortcomings, the better I could cover my weaknesses with the robot's behavior.
>
> **Q10:** Do you think the robot helped you or blocked you from doing your task?
>
> **A:** It's absolutely helping. I just needed to learn its behavior, then I could act in a way to help the robot understand my goal and help me achieve it.

These observations highlight the potential benefits of shared control systems in improving human performance by utilizing robotic support and adaptability.

3.5.4 Discussion

In shared control systems, determining control allocation—who should be in control, when, and to what extent—depends on the task's properties, the capabilities of the human and robot, and their ability to achieve a common goal. In scenarios where the human excels in one aspect and the robot in another, a continuous weighted control approach may not be optimal. Instead, switching control, which alternates control between human and robot, might be more effective. Alternatively, humans may adjust the control weighting manually, seeking more assistance from the robot or taking full control as needed.

This dynamic exchange allows both human and robot to learn from each other's strategies. In the experiment, humans initially explored robot behavior, leading to higher performance levels than achievable alone.

3.6 Conclusion

This chapter explored the human adaptation process to human-robot shared control systems. Two control scenarios were investigated: a human-in-the-loop controller, where the robot remains passive, and a human-robot shared controller, where the robot actively predicts and assists based on inferred human intentions. The intention inference mechanism was pivotal for aligning both agents towards a common goal. Human subjects performed the task under both conditions in multiple trials over four days to examine learning progress and adaptation.

The results indicated that humans adapted to the system and learned the task more rapidly when assisted by a robot, achieving higher levels of performance. This finding highlights the potential for synergistic human-robot collaboration to improve task performance and faster adaptation. The active role of the robot significantly influenced human actions in the shared control scenario.

The shared control framework appears to facilitate greater individual improvement for the human operator compared to passive control of the robot. This improvement is likely due to the increased engagement of the human operator with the system, as shared control encourages active learning by requiring the human to understand, predict, and adapt to robot behavior. By collaborating with the robot, the operator develops a deeper understanding of the capabilities and limitations of the system. This active participation not only enhances the operator's decision-making and motor skills but also allows them to leverage the robot's strengths more effectively, leading to a more refined and efficient control over time.

CHAPTER IV

ADAPTIVE BLENDED HUMAN-ROBOT SHARED CONTROL

4.1 Introduction

Building upon the findings of the previous chapter, which highlighted the benefits of human adaptation in shared control systems, this chapter investigates how adaptive mechanisms can enhance human-robot collaboration. In Chapter III, we explore how human subjects interacting with shared control systems showed faster learning and improved task performance when assisted by robots. The shared control approach allowed humans to leverage the robot's strengths while adapting to its behavior, resulting in a more refined control process over time.

This chapter extends that exploration by examining how adaptive blending strategies in shared control systems can further optimize human-robot collaboration. Although shared control has proven to facilitate faster adaptation and learning, the blending of human and robot inputs remains a crucial factor in determining the overall effectiveness of the system. By dynamically adjusting the blending parameter based on the evolving context, human input, and task requirements, adaptive shared control systems can create a more balanced and responsive interaction, enhancing both performance and user experience.

The focus here is on developing an adaptive blended human-robot shared control framework that intelligently combines the inputs of both parties. As autonomous systems excel in precision and consistency and humans bring flexibility and intuition, blending their inputs in a way that adjusts to the context and human intention becomes a key strategy for enhancing robustness and accuracy. We aim to investigate

how we can dynamically align human-robot control toward shared goals. This approach not only helps mitigate human limitations such as fatigue or decision errors but also ensures that the robot can respond effectively to new patterns or unexpected changes in the environment.

Humans and autonomous systems offer unique strengths that align with specific tasks. Autonomous systems excel at executing precise and rapid movements, particularly in repetitive tasks with clearly defined objectives [87]. In contrast, human performance can be error prone, especially in complex or prolonged tasks, where factors such as fatigue and monotony play a role [88]. However, humans possess the unique ability to recognize and contextualize new patterns, apply inductive reasoning, and quickly adapt to unexpected situations. Autonomous systems, on the other hand, often face challenges in effectively handling unforeseen circumstances [89]. The key driver behind developing human-robot shared control systems is to take advantage of the strengths of both humans and robots, creating a synergistic partnership that improves overall control performance in terms of robustness and accuracy while alleviating the workload on human operators [58, 59].

In human-robot shared control systems, a widely adopted strategy is blended shared control, in which the human and autonomous system commands are continuously integrated by using a blending parameter [26, 27, 90, 91]. The blending parameter can be manually or automatically set, although determining the optimal value is not always straightforward. In the work presented in [92] a method is proposed for setting the blending parameter based on a function that assesses the disparity between autonomous control and human control commands. Another perspective, proposed in [93], suggests setting the blending parameter according to the probability calculated from a finite state machine, which quantifies the sub-goal transition probabilities within a given construction task. In addition, in [94] a fusion method is proposed that calculates and combines the input commands with respect to the

admissibility or correctness of the control commands of each party.

A key element for shared control scenarios is the estimation of human intention. In a shared control framework, enabling robots to learn the human intention is a challenging but critical issue to address. Considering that human actions are often goal-directed [37], effective collaboration between humans and robots requires that both parties work toward the same goal. Nevertheless, achieving precise predictions of human intention under all circumstances appears challenging due to the inherently intricate and sometimes stochastic nature of human behavior. Not only can different human operators employ varying strategies to accomplish a given task, but even the same human operator may employ distinct approaches when repeating the same task. In addition, the level of human expertise adds another level of complication to the intention estimation problem, as humans adapt their movement patterns through practice [38].

Recently, in adaptive shared control frameworks, recent studies have focused on a more effective and safer human-robot interaction. In [95] a prospective method based on heuristics is proposed that adapts the triplet Competence-Availability-Possibility-to-act (CAP) is proposed to discover possible conflicts of shared control between humans and the autonomous system. In [96] an indirect shared control method is proposed to model the situation where the autonomous agent and the human have different intentions, which can cause severe conflicts during cooperation, using a Nash equilibria strategy the best response for the automated system is obtained, which can guide the automated controller to act more safely and comfortably. In another study [97] inspired by the sensorimotor mechanisms of the primate brain, object affordances were utilized for both intention estimation and task execution, to generate an altruistic model. To assess how human partners interact with such an altruistic model, a study was conducted with naive human subjects, which resulted in human partners being more involved in exploiting behavior.

4.2 Adaptive Blended Shared Control Framework

In this section, the description of the proposed adaptive blended shared control framework and its components is presented. The general framework is illustrated in Figure 9. Here, the human and autonomous system are envisioned to collaborate and interact congruently in a perception-action cycle to perform a given task through a robotic manipulator. This framework has the potential to be used where full comprehensive sensing of the environment is impractical and where quick decision-making ability of the human when unseen or unexpected events may occur is needed to provide a level of robustness that is difficult to encode. Here, the proposed adaptive blended shared control framework is introduced, and its components are explained from a general perspective. The system is composed of three main components, namely, Human Control, Robot Control and Shared Control, which are described in the following subsections.

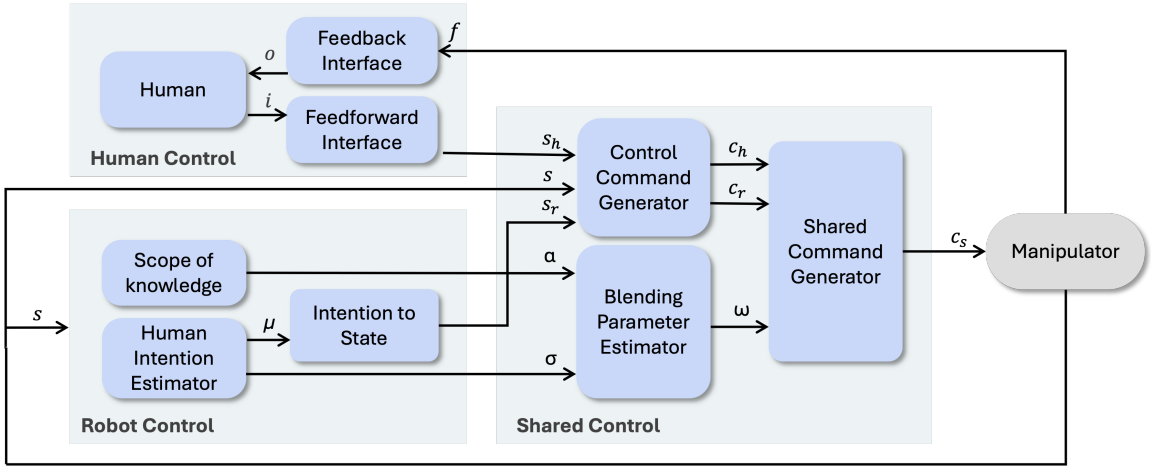


Figure 9: The proposed adaptive blended shared control framework, where the human and the robot collaborate to perform a given task through a manipulator

4.2.1 Human Control

In this shared control framework, human remains in control for decision making, dealing with unseen situations, and finding solutions for these matters rather than

being actively and continuously in control. The interaction between humans and the shared control system is effectively managed through the utilization of feedforward and feedback interfaces. These interfaces serve as channels for conveying information, instructions, and responses between the human operator and the system.

4.2.1.1 Feedforward Interface

To send the human input in real-time to the shared control system, we provide the human with a feedforward interface. In this framework, human operator input (noted i in Figure 9) is taken through an interface, and based on that given input, the human desired next task state; refers to the specific state or condition that the human operator wants the system to achieve or move towards; is obtained (noted with s_h in Figure 9) which is then fed to the shared control system to control the robotic manipulator. There are a number of different interfaces that can be used here; however, the type of interface used is highly dependent on the application and the task to be performed. Some possible examples for this feedforward interface include keyboard, mouse, joystick, touchscreens, audio, motion tracking systems, and haptic devices.

4.2.1.2 Feedback Interface

To provide sufficient information to the human who might be less aware of what is happening in the situation where the control is shared with a robot, we consider a feedback interface. In this framework, the feedback interface is responsible for communicating feedback information (noted with f in Figure 9) from the system to the human operator as their observation of the state of the task (noted with o in Figure 9). The feedback interface commonly takes two forms: visual feedback, often presented either directly or through camera recordings, and haptic feedback. Given that teleoperation tasks involve physical interactions, leveraging the sense of touch can be especially effective in providing force or haptic feedback to the human operator.

4.2.2 Robot Control

The Robot Control in the proposed adaptive blended shared control framework has three modules, namely, Scope of Knowledge, Human Intention Estimator, and Intention to State modules.

4.2.2.1 Human Intention Estimator

In this framework, the Human Intention Estimator observes the current state of the task (marked with s in Figure 9) and estimates the human intention. Here, human intention refers to the trajectory that the human operator will take in the subsequent time steps to accomplish the given task. Note that we consider that a task can be accomplished in different ways (for instance, there could be different goals that would satisfy task completion). The human intention is noted with μ in Figure 9. The human intention estimator module also provides the standard deviation of the human intention estimation indicated by σ in Figure 9.

4.2.2.2 Scope of Knowledge

This module addresses the problem of deciding what the robot implicitly knows or what it can derive from its knowledge. Addressing this problem in this framework is important as the robot's estimation of human intention would not be valid when the observations are out of the robot's scope of knowledge. This module takes the current state (s in Figure 9) and provides a signal (α in Figure 9) indicating whether the given state is seen by the robot before or not.

4.2.2.3 Intention to State

This module interprets the estimated human intention. Although the estimated intention and the information we aim to measure (typically, the next task state) may not be identical, they are related. As a result, the robot maps its estimation of human intention to the next state of task, denoted as s_r in Figure 9.

4.2.3 Shared Control

The Shared Control component is responsible for taking the autonomous command and human command and blending them. This component is adaptive and able to give the appropriate portion of the control at the right time to the right party while keeping the task trajectory smooth and seamless. In this framework, the Shared Control component has three modules, namely Control Command Generator, Blending Parameter Estimator, and Shared Command Generator.

4.2.3.1 Control Command Generator

The control command generator takes the next desired human state s_h, and robot state s_r, and generates their corresponding control commands c_h and c_r given the current state of the task s. Safety measures of control commands can be applied in this module as well, for instance, human control commands c_h and c_a can be filtered based on the constraints of the system and manipulator.

4.2.3.2 Blending Parameter Estimator

Estimation of blending parameter is based on two parameters; (i) α that indicates if the observed state is known in the robot's scope of knowledge, and (ii) σ that depicts the standard deviation of human intention estimation. Given α and σ, the mixing parameter ω is estimated that indicates the human share in control. If $\alpha = 1$ then σ is used to calculate a confidence factor and obtain the blending parameter ω, otherwise the robot's estimation of human intention will be marked as invalid, and the shared control system gives all the weight of control to the human operator, i.e., ω is set to 1.

4.2.3.3 Shared Command Generator

Shared control command c_s is generated based on human control command c_h, robot control command c_r and the blending parameter $0 \leq \omega \leq 1$. In the proposed framework as in Figure 9, the convex combination of human control command and autonomous control command is used to obtain the shared command, as shown in equation (12), where ω is dynamically adjusted.

$$c_s = \omega c_h + (1 - \omega) c_r \tag{12}$$

4.3 Human to Robot Skill Transfer

For a robot to act as a shared control partner, it must be equipped with the skill of autonomously performing the task. For this purpose, the learning from demonstration (LfD) technique [98] is used, and task execution data is collected via teleoperation. To teach the tasks to the robot based on the collected data, the Conditional Neural Movement Primitives (CNMPs) [2] are extended and used to learn movement trajectory distributions corresponding to the learned tasks. CNMPs are originally used for learning from demonstrations and are built on top of Conditional Neural Processes (CNPs) [1]. After learning, CNMPs can be conditioned on single or multiple desired trajectory points at specific time steps to produce trajectory distributions that satisfy the given conditions. Since CNMP work is formulated for learning from perfect demonstrations, the demonstration data quality directly affects the estimations. On the other hand, when various demonstrations are given to the network for learning, it becomes evident that certain demonstrations are more effective than others, and each demonstration may result in a varying level of performance. In the original formulation of CNMPs, given a condition, the network aims to generate a trajectory that satisfies the corresponding condition without considering the performance factor. Unlike the original CNMPs, in cases where different demonstrations have different levels of performance, the performance factor is proposed to be taken into account, and the

network estimation is led towards trajectories that exhibit a superior performance level while ensuring that the given conditions are still satisfied, and this is called the weighted-CNMP model.

4.3.1 Learning from weighted demonstrations

For weighted-CNMP, it is essential that the provided demonstrations are accompanied by an associated evaluation score. These scores are obtained from evaluating the demonstrations across various task-specific factors. Subsequently, these evaluation scores are used as the basis for determining the impact of each demonstration on the network estimations, effectively assigning weights based on their quality. Using the evaluation score, a normal distribution is formed, centered around the demonstrations with higher evaluation scores [56]. Training is initiated by selecting a demonstration using the formed normal distribution. From the selected demonstration, some observation and query points are sampled randomly, and the mean μ and standard deviation σ at the query point are then estimated by the network. With the ground truth of the query, the loss value is calculated and minimized by gradient descent. The algorithm for training weighted-CNMP is provided in Algorithm 1.

To examine the impact of weighting the demonstrations and to compare the outputs of CNMP with the modified network, three movement trajectories were generated as shown in Figure 10. Two models were designed to learn and generate these trajectories from a start point to a target point while avoiding the obstacles on the way. The trajectories with a longer distance to the obstacles are more desired and thus were given higher scores. After the models were trained, trajectories were generated by both models given one single observation as a condition point. It was expected that a trajectory satisfying the given condition would be generated by both models. Figure 10 shows the trajectories generated by these two models. The results show

Algorithm 1 Weighted-CNMP training for learning demonstrations as trajectory distributions

Demonstrations ← Load the demonstrations
Scores ← Evaluation(demonstrations)
Pd ← NormalDistribution(demonstrations, scores)
Initialize Lt ▷ Lt is the $loss$ threshold
Initialize m ▷ m is the maximum number of observations from one demonstration
while $loss \leq Lt$ **do**
    A demonstration D is selected using Pd
    $n = randint(m)$
    n observation points are picked from D in the form of $[t_i; f(t_i)]$ ▷ $f(t_i)$ is the features of demonstration at time t_i
    A query is picked in the form of $[t_q; f(t_q)]$
    (μ_q, σ_q) of $f(t_i)$, for t_q, is estimated
    $loss = -logP(f(t_q)|\mu_q, softmax(\sigma_q))$ ▷ Having the ground truth of the query, the loss value is calculated
    Stochastic gradient is applied to minimize $loss$
end while

that, given a common condition (where all the given demonstrations satisfy that condition), an average trajectory is generated by the CNMP model, while a more desired trajectory is generated by the weighted model.

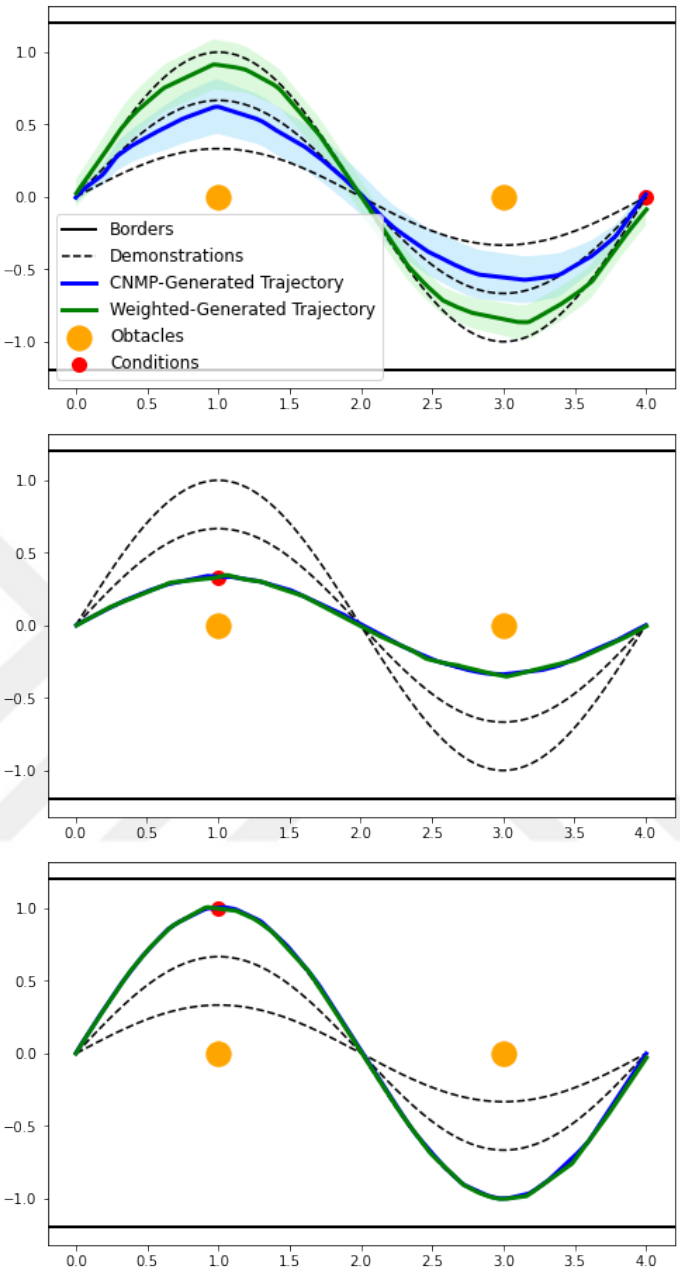


Figure 10: Generated trajectory; x(m) and y(m) position; by CNMP (blue line) vs weighted model (green line). The light blue and light green color show the standard deviation of the estimation. top figure shows the generated trajectories when being conditioned on a common point, which shows when both models are uncertain, the weighted model generates a more desired trajectory.In middle and bottom figures, models are conditioned on an uncommon point (satisfying only one seen trajectory) and generate similar trajectories.

4.4 Human Intention Estimation

The aforementioned neural network model (Weighted-CNMP) is proposed to be used for human intention estimation by being directly conditioned in real-time based on human behavior and the current state of the task. In this approach, the neural network leverages the dynamic interaction between the human operator and the system, allowing for a seamless integration of human intentions into the control process.

Consider a scenario in which the manipulator is started to be moved by the human operator to a desired position and orientation at time t. The neural network model, having already been trained on various task conditions and demonstrations, is utilized in this context to interpret and predict the human operator's intended actions. As the model is conditioned on the current state of the task, a trajectory that satisfies the given condition for accomplishing the task is generated. This generated trajectory is considered as the estimate of human intention, providing insights into the human operator's goals and next expected movements.

The estimate of human intention μ and the variance of the estimation σ^2 are obtained by the neural network, offering a quantitative measure of the predicted trajectory's precision and reliability. Based on these estimations, the autonomous control commands are generated, enabling the system to effectively collaborate with the human operator. This approach ensures that the autonomous system can adapt to the operator's real-time input, enhancing the overall efficiency and safety of the task execution.

4.4.1 Adaptive shared control policy based on human intention

With the human and autonomous control commands available, the adaptive shared controller is tasked with blending these commands and driving the manipulator. When human intention is estimated, the human's desired trajectory, μ, and the variance of the estimation, σ^2, are generated.

At each time step of the task, t, the values of $\mu_t(i = t+1 : T)$ and $\sigma_t^2(i = t+1 : T)$ are estimated for the subsequent time steps ($i = t + 1 : T$), where T is the final time step of the task. The mean of the estimation's variance, beginning at time step $t + 1$ and continuing until the task completion time T, is denoted as $\sigma_t^2(i = t + 1 : T)$ and is used to define a confidence factor, c_f. The confidence factor, $c_f(t)$, is calculated using equation (13) and indicates the level of confidence the autonomous controller has in its estimation of the human intention at time t.

$$c_f(t) = 1/(K\frac{\sum_{i=t+1}^{T}\sigma_t^2(i)}{T-t}) \tag{13}$$

Where K is a scalar that was tuned manually.

When the value of the confidence factor, $c_f(t)$, is high, indicating that the autonomous controller has a high level of confidence in its estimation of human intention, it becomes possible to reduce the human effort required. In such cases, the autonomous controller can assume a larger portion of control. To achieve this, the weight of the human control command is determined using:

$$w(t) = 1 - c_f(t), \quad \text{where} \quad 0 \leq w(t) \leq 1 \tag{14}$$

where $w(t)$ represents the adaptive blending parameter that signifies the proportion of control exerted by the human command at time t.

The blending parameter $w(t)$ is sensitive to the value of K, which is adjusted based on the variance, σ^2, of the estimation. Specifically, K is tuned so that the confidence factor $c_f(t)$ approaches 0 when σ^2 reaches its maximum value. This ensures that when the estimation's variance is high and the confidence is low, the human control command is given more weight, and the autonomous control command is reduced.

The final shared control command, which directs the manipulator at time t, is calculated using the following equation:

$$C_{sh}(t) = w(t)C_h(t) + (1 - w(t))C_r(t) \tag{15}$$

In this equation, $C_{sh}(t)$ represents the shared command, $C_h(t)$ denotes the human command, and $C_r(t)$ indicates the autonomous command at time step t. By adjusting the weight $w(t)$ based on the confidence factor, the shared control command can be dynamically blended to balance human and autonomous contributions effectively.

4.5 Framework Assessment

4.5.1 Human intention estimation assessment

To examine the human intention estimator proposed in adaptive blended shared control framework, a task was designed and simulated in CoppeliaSim Edu [99]. In this task, a mobile robot (Pioneer p3dx) was required to reach a given target point within a fixed time of 15 seconds while avoiding obstacles along the way. The simulation scene is shown in Figure 11.

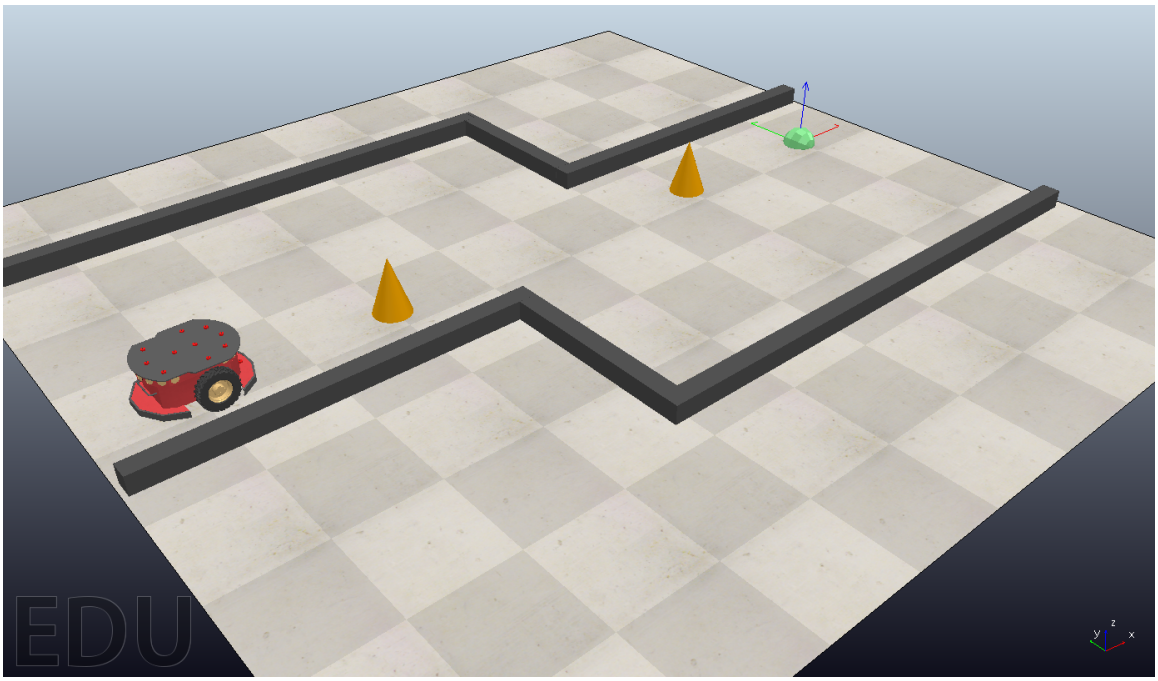


Figure 11: Mobile robot simulation scene

For this task, human input was taken via a computer mouse as the feedforward interface. Mouse displacements in the x and y axes were used to calculate the linear and angular velocities of the mobile robot. The obtained velocities were then converted into commands for the right and left wheel motors. Visual feedback was provided to the human operator through the computer screen.

The weighted-CNMP model within the autonomous controller was initially trained using a fixed set of demonstrations (three in this experiment), as illustrated in Figure 10.

Evaluations corresponding to these demonstrations were provided to the learning algorithm, as detailed in Algorithm 1. These evaluations measured the effectiveness of the trajectories in avoiding obstacles by monitoring their closest distances to the obstacles. The network was designed to predict the positions x and y of the mobile robot based on one or more observations. Observations were inputted into the network as tuples of the form $o = [t, x(t), y(t)]$, and the network then estimated the mean and variance of $x(q)$ and $y(q)$ for any query point. For this experiment, the queries span the entire range of time steps ($q = 0 : T$).

Given that human behavior may vary over time, the intention estimation was performed continuously at each time step. μ_t represents the estimated human intention at time t, and $\mu_t(t+1)$ indicates the desired position of the mobile robot at the subsequent time step $t+1$. Using the confidence factor $c_f(t)$ and blending parameter $w(t)$, which are calculated according to equations (13) and (14), the shared command C_{sh} was determined using equation (15). The process for this shared control loop is outlined in Algorithm 2.

Using this framework, the task in the simulated environment was accomplished through collaboration between a human operator and the autonomous system. The performance of the intention estimation method is shown in Figure 12. Additionally, the blending parameter, human and autonomous commands, and task state for the

Algorithm 2 Shared Control Loop

Model $\leftarrow$ Load the trained model
Initialize K
while $t \leq T$ **do**
    $R_x, R_y \leftarrow$ Read robot state
    $C_h \leftarrow$ GetHumanCommand(Interface)
    condition = $[t\ ,\ R_x\ ,\ R_y]$
    $\mu, \sigma \leftarrow$ Model(condition)
    $C_a \leftarrow$ GetAutoCommand(μ)
    $c_f \leftarrow$ Confidence(σ, K)
    $w \leftarrow$ GetBlendingParameter(c_f)
    $C_{sh} = wC_h + (1 - w)C_a$
    Send the C_{sh} to robot
end while

entire duration of the task are depicted in Figure 13. As illustrated in Figure 13, the blending parameter, which indicates the human's share of control, increases at certain common points of the task (such as the beginning and between obstacles) because the autonomous system cannot estimate human intention with high confidence at these points. This is due to the fact that different trajectories, which match these common points, were demonstrated to the autonomous system during training.

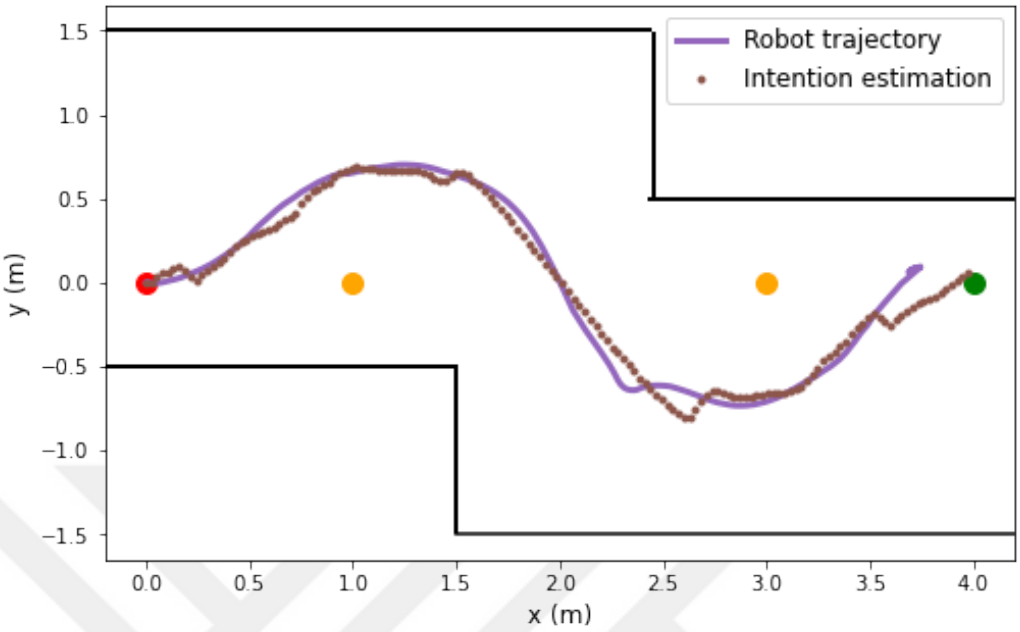


Figure 12: Mobile robot trajectory and the intention estimation. In here the human intention refers to estimated desired robot position at the next time steps. Red, orange and green dots are starting point, obstacles, and target point respectively.

4.5.1.1 Teaching shared trajectories to robot

When this framework was used to perform a task, new trajectories were formed through the shared control established by the combined human and autonomous control commands. These trajectories could be beneficial in making the autonomous system more adapted to the human operator, and vice versa, potentially leading to stronger collaboration. To evaluate this idea, two setups were designed. In the first setup, three shared demonstrations with evaluations were added to the demonstration data set, and in the second setup, three pure human demonstrations were added to the demonstration data with evaluations.

With these two data sets, two new models were trained. To compare the performance of these models, a human operator tested each model 10 times to accomplish the designed task. The sum of the absolute difference between human commands and autonomous commands was considered a metric of the effectiveness of collaboration.

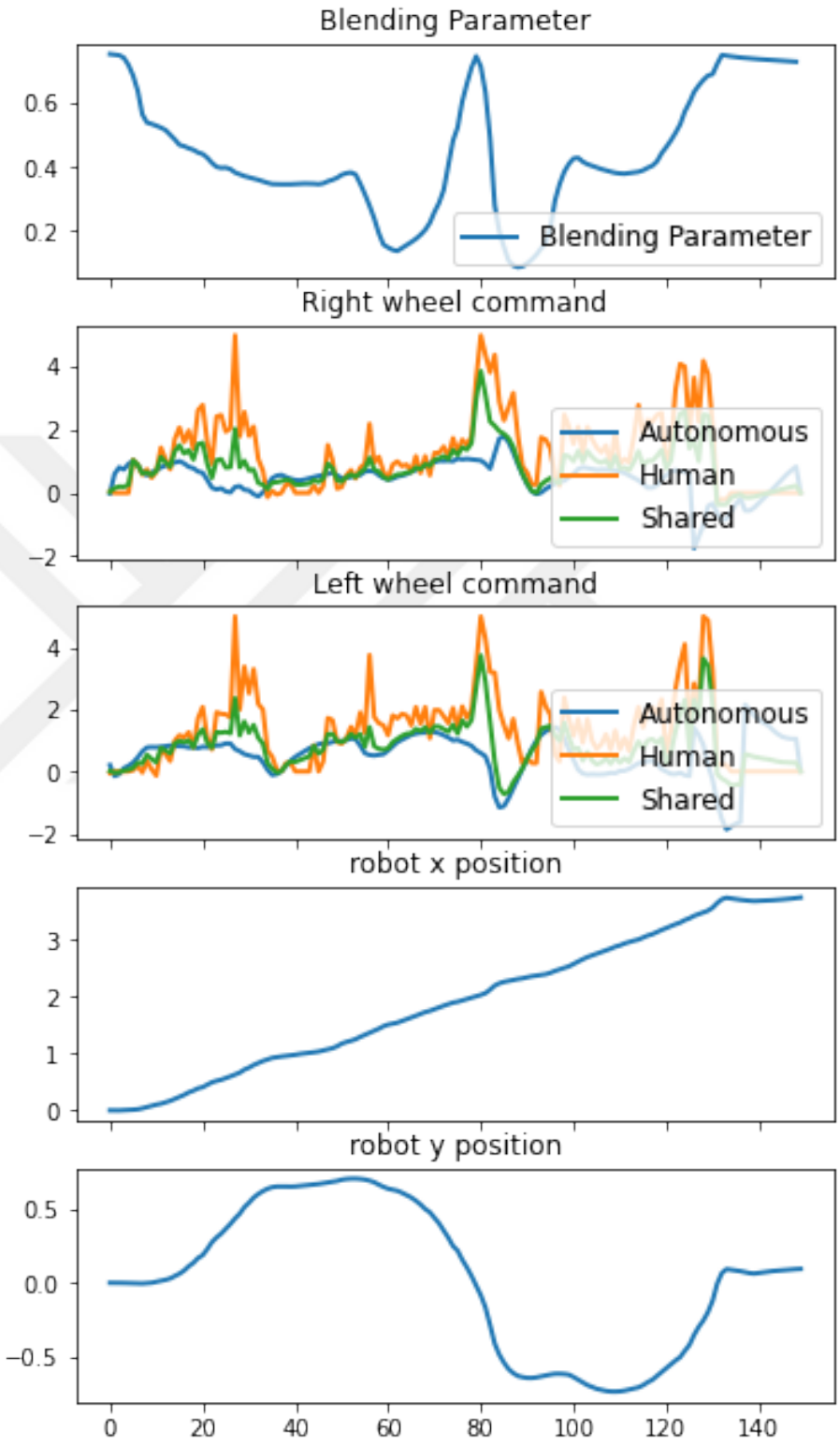


Figure 13: Blending parameter, human and autonomous commands for right and left wheel of the mobile robot, and task state (robot's position). The blending parameter that denotes the human share in control increases where the mobile robot is in some common condition points of the task (such as the beginning and in between the obstacles)

As the blending parameter depicts the human share in control, it was chosen as the metric to show human effort in collaboration. The means of these two metrics were calculated and compared in 10 trials for each setup (see Figure 14). The comparison indicates that the difference between the human and autonomous commands was lower in Setup 1. Furthermore, human effort was slightly less in Setup 1 compared to Setup 2. Overall, these data suggest that adding shared trajectories to the knowledge of the autonomous system induces better collaboration that can also reduce human effort.

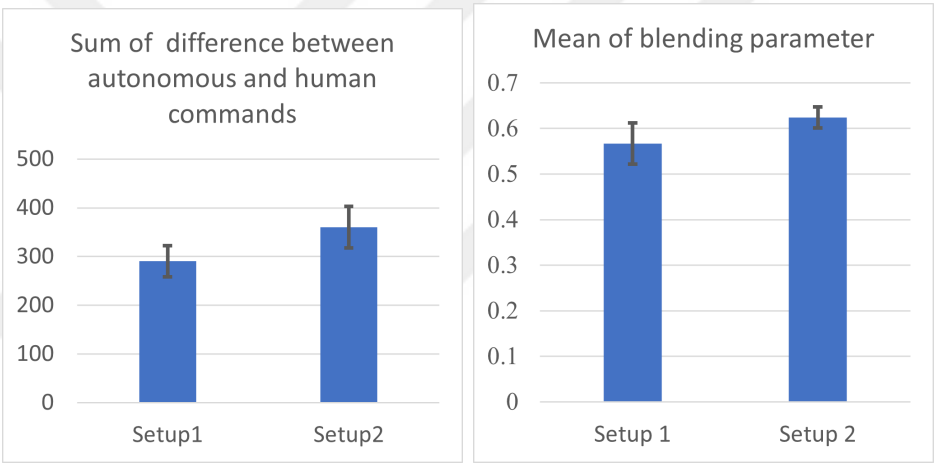


Figure 14: Sum of absolute difference between human and autonomous commands, and mean of blending parameters in 10 trials for each setup. Error bars are standard deviations.

4.5.2 Scope of Knowledge Assessment

To assess the efficacy of the Scope of Knowledge module in the proposed adaptive blended shared control framework, an obstacle avoidance task in simulation was designed.

In this task, a 7-degree-of-freedom (DOF) robot manipulator, the Franka Emika Panda, equipped with a pole attachment, was used in the Coppeliasim [100] environment. The simulated environment included three target goals and five obstacles, as visually depicted in Figure 15. The objective of this task was for the robot's end

effector to be efficiently navigated to one of the predefined target goals while avoiding the obstacles.

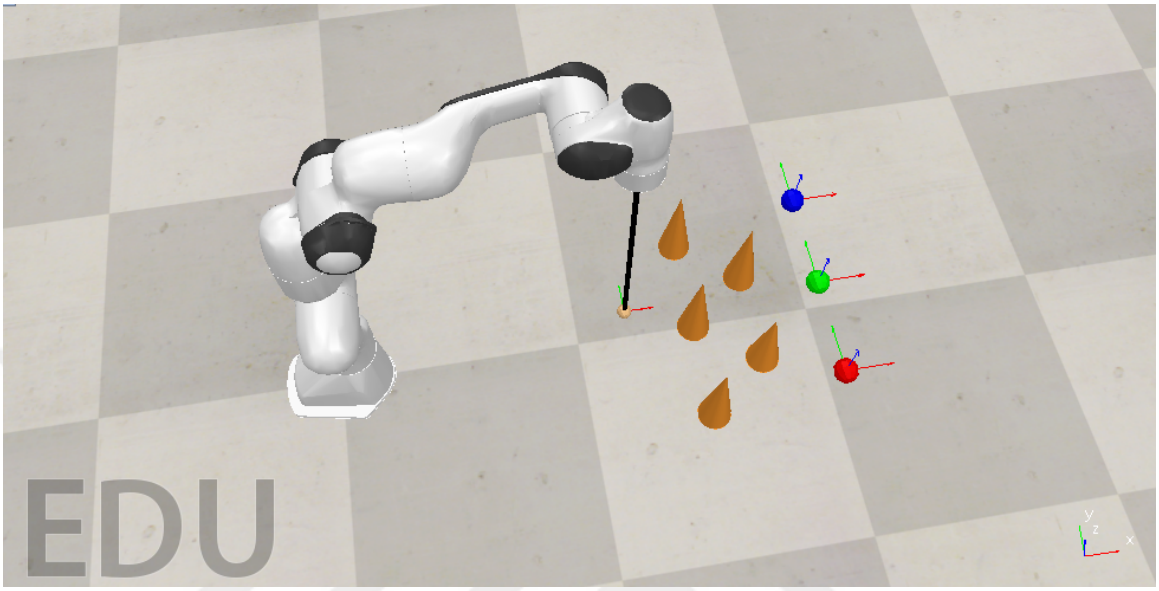


Figure 15: Simulated scene for the obstacle avoidance task in Coppeliasim

4.5.2.1 Teleoperation

Teleoperation of this task was carried out using a computer mouse. The displacement of the mouse along the x and y axes was adjusted for sensitivity and then used to move the end-effector of the robotic arm to the desired position (with the z-axis position fixed in this experiment).

4.5.2.2 Collecting Demonstrations

Trajectory demonstrations were collected as the robot was teleoperated by a human to reach the task objective. Four demonstrations for each goal were recorded, resulting in a total of 12 demonstrations. Data were collected at a frequency of 20 Hz. The demonstrations were evaluated based on the mean minimum distance to the closest obstacle recorded during each step of task execution. These evaluation scores were used to weigh the collected demonstrations for training. The collected demonstrations were then pre-processed by smoothing and re-sampling to reduce noise and

ensure all demonstrations were of the same length (250 data points in this experiment, corresponding to the longest demonstrated trajectory length).

4.5.2.3 Skill transfer to robot

The robot was trained to learn the trajectory of the end effector using Algorithm 1. After training, a trajectory can be generated by conditioning the weighted-CNMP on any desired end-effector position. Figure 16 demonstrates the demonstrated trajectories for training the model and also the generated trajectories by the trained model when conditioned at different points. You may also observe the video of the robot in action, running the generated trajectories when conditioned on different points via the following link https://youtu.be/3oDurufY9rI.

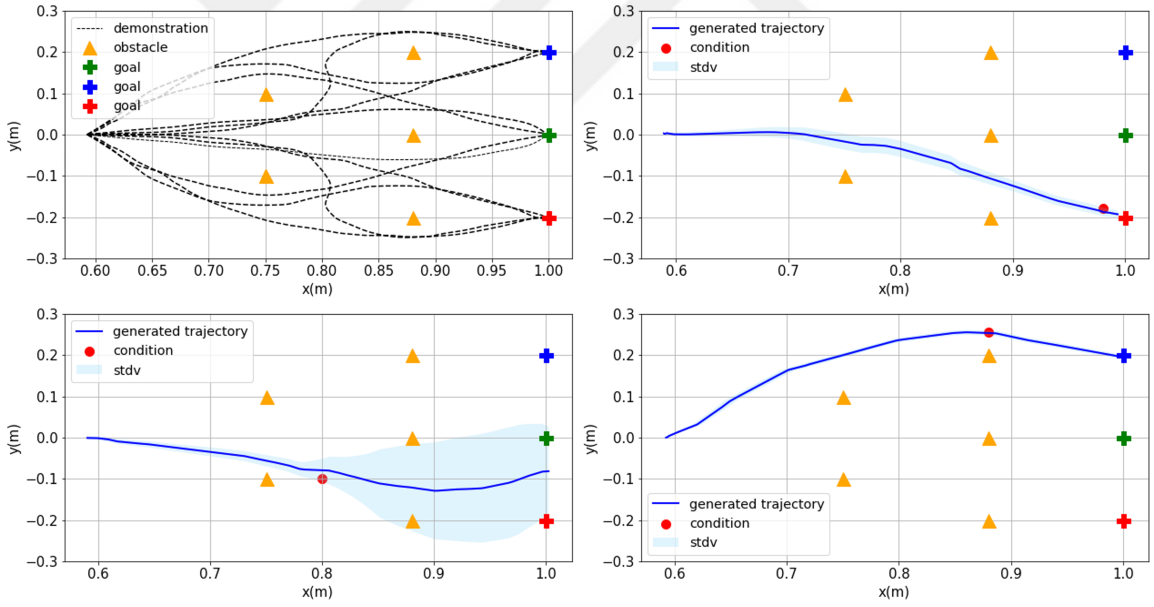


Figure 16: Demonstrated trajectories for training the Weighted-CNMP, and the estimated trajectories after training when being conditioned for obstacle avoidance task. Note that the standard deviation of estimation is high when the given condition satisfies multiple trajectories for reaching different goals.

4.5.2.4 Learning Scope of Knowledge

The state of the task was defined as the position of the robot's end-effector. The states observed in the demonstrations used for training were labeled as "known." To

enhance generalization, the training data, which included 3,000 observed states (from 12 demonstrations, each with a length of 250), were augmented with 17,000 additional states. These additional states were randomly generated within the minimum and maximum values of the state parameters and labeled based on their Euclidean distance to the known states. A threshold of $1cm$ was established for this distance, adjustable to meet the specific precision required for the execution of the task. The labeled data were used to train and test the MLP classifier. The dataset was partitioned into two subsets: a training set comprising 80% of the data and a test set containing the remaining 20%. The performance of the classifier was evaluated on both sets, achieving a precision of 0.9646 on the training data and a comparable precision of 0.9642 on the test data. Figure 17 illustrates the robot's scope of knowledge and the performance of the MLP classifier.

4.5.3 User Study

For this experiment, two setups were designed. In the first setup, the framework shown in Figure 9 was applied. In the second setup, the robot's scope of knowledge component was removed, meaning that the robot's estimation of human intention was always considered valid. The task was initiated by the human operator's first input through the feedforward interface, while the robot started the estimation of human intention to obtain the next state of the task with a confidence factor. In the first setup, the blending parameter was calculated based on both the confidence factor and the output of the robot's scope of knowledge. In the second setup, the blending parameter was calculated solely based on the confidence factor. The target goal of the human was unknown to the robot and was adaptively estimated at each time step.

The performance factors considered for this experiment included completion time, mean distance to the closest obstacle, and the blending parameter indicating the

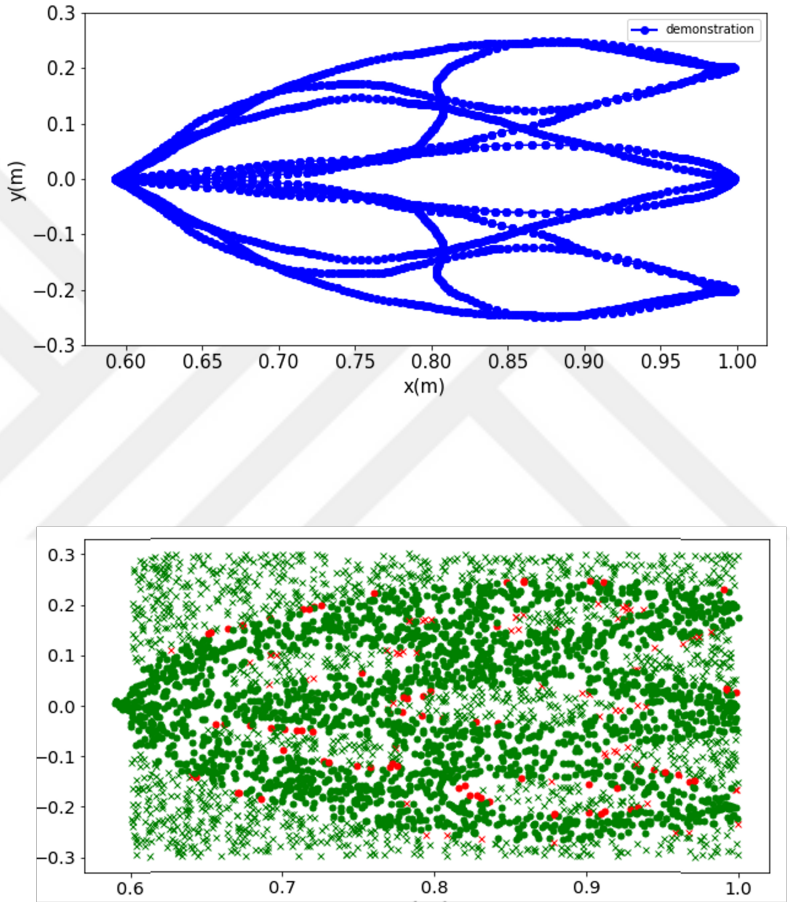


Figure 17: Representation of robot's scope of knowledge -In the top row, the states demonstrated to the robot during training are shown. The row below illustrates the results of state classification, with the symbols "o" and "x" indicating states classified as "known" and "unknown," respectively. States correctly classified are denoted in green, while misclassified states are highlighted in red.

Table 5: Outcome comparison between setup 1 and setup 2 for reaching with obstacle avoidance task for all trials in each setup. N=144

	Setup 1		Setup 2	
Outcome	Frequency	Percent	Frequency	Percent
Success	132	91.7	119	82.6
Failed-O	2	1.4	7	4.9
Failed-G	10	6.9	18	12.5

human share in control. A trial was defined as successful if the desired target goal was reached without hitting any obstacles.

A within-subjects user study was conducted with 16 naive subjects (4 females and 12 males between the ages of 21 and 35). Each subject performed the task for each setup 9 times. There were 3 target goals for each setup, and for each target goal, 3 trials were recorded. To decrease the learning effect, the order of target goals was randomly shuffled and each subject switched between the setups after each trial. A total of 144 trials for each setup were collected. The results of these trials were grouped into three categories: (i) 'Success', marking a successful trial; (ii) 'Failed-O', marking a failed trial that occurred by hitting the obstacles; and (iii) 'Failed-G', marking a failed trial due to not reaching the desired target goal. The outcome of this experiment is shown in Table 5, indicating a substantial difference in success rates: 91.7% for Setup 1 and 82.6% for Setup 2.

To assess whether there was any statistically significant difference in the performance factors between these two setups, Wilks' Lambda test was conducted. This statistical test is used in repeated measures multivariate analysis of variance (MANOVA) to assess the significance of differences between groups when there are multiple dependent variables [101, 102]. This test was chosen because it can handle situations where the dependent variables are correlated or have different scales of measurement. It is also robust to violations of normality when the sample size is large and the group

Table 6: Descriptive statistics of within-subject factors among the successful trials in both setups for obstacle avoidance task, N=109

	Setup 1		Setup 2	
Measures	Mean	std. dev	Mean	std. dev
Time (s)	4.347	1.745	4.878	2.334
Distance (m)	0.104	0.023	0.097	0.011
Blending Parameter	0.624	0.105	0.532	0.056

Table 7: Repeated measures multivariate analysis of variance for within-subjects setup, Wilk's lambda test results for the successful trials for obstacle avoidance. N=109

	Value	F	Sig.
Wilk's Lambda Test	0.534	30.847	< 0.001

sizes are balanced.

To compare these two groups, the failed trials and their paired trial in the other setup were removed from the samples. Thus, the test was conducted on the successful trials of both setups, corresponding to 109 trials from each setup. The within-subjects measures in this test were the completion time, the distance to the closest obstacle, and the blending parameter. The descriptive analysis of these within-subject factors is given in Table 6, and the result of the repeated measures multivariate analysis of variance is given in Table 7. As shown in Tables 6 and 7, the value of Sig. (P-value) is less than 0.05, which allows for the rejection of the null hypothesis, concluding a significant difference between the two setups: Setup 1, where robot autonomy was restricted to its scope of knowledge, had a shorter completion time, a greater distance to the closest obstacle, and a higher blending parameter.

4.6 Discussion

The experiment aimed to evaluate the performance of the adaptive blended shared control framework in two different setups: one with the robot's scope of knowledge component and one without it. The results demonstrate a significant advantage of the first setup, where the robot's autonomy was constrained by its scope of knowledge. This setup achieved a higher success rate of 91.7% compared to 82.6% in the second setup. The lower frequency of failures in the first setup, particularly those categorized as "Failed-O" (collisions with obstacles), highlights the effectiveness of incorporating the robot's knowledge component in ensuring safer navigation.

The performance factors analyzed, including completion time, mean distance to the closest obstacle, and blending parameter, further support these findings. The first setup not only resulted in shorter completion times but also maintained a greater distance to obstacles, indicating improved efficiency and safety. The higher blending parameter observed in the first setup suggests a greater reliance on human input, which may have contributed to the system's adaptability in navigating complex environments.

The within-subjects design and the use of Wilks' Lambda test provide robust statistical evidence supporting the significant difference between the two setups. The chosen method effectively handled the dependent variables correlated and ensured a balanced analysis of the performance factors across different subjects.

The integration of the robot's scope of knowledge allowed for more informed decision making during the shared control process, facilitating better cooperation between the human operator and the robot. This adaptability may prove crucial in real-world applications where environments and tasks are dynamic and unpredictable.

4.7 Conclusion

This study demonstrates the importance of incorporating a robot's scope of knowledge in a shared control framework to enhance task performance and safety. The first setup, which utilized the robot's knowledge to validate its estimation of human intention, showed superior results in terms of success rate, completion time, and obstacle avoidance. The findings points out the value of adaptive control systems that can leverage both human expertise and autonomous capabilities.

Future research should explore the application of this framework in more complex and varied task environments. Additionally, further studies could investigate the long-term effects of such adaptive systems on human-robot collaboration and the potential for reducing operator fatigue and improving task efficiency over extended periods. The continued development of intelligent shared control systems holds great promise in advancing the field of human-robot interaction and expanding the capabilities of robotic systems in diverse settings.

CHAPTER V

SHARED CONTROL IN MEDICAL SUTURING

5.1 Introduction

Building upon the insights from Chapter IV, where we explored the benefits of incorporating a robot's scope of knowledge into shared control frameworks, this chapter shifts focus to a real-world application: medical suturing. In Chapter IV, the findings highlighted how shared control systems, when adaptive and informed by both human expertise and robotic knowledge, can significantly enhance task performance and safety. These insights are particularly relevant to the medical field, where precision, safety, and efficiency are paramount.

In the medical domain, especially in procedures like suturing, the collaboration between humans and robots through shared control systems can transform the way surgeries are performed. The robot's ability to validate its actions based on human input, much like in the previous experiments, can contribute to higher success rates and improved safety during delicate medical operations. This chapter explores how shared control frameworks can be adapted to support medical tasks, specifically suturing, where both robotic precision and human decision-making play crucial roles.

As automation in healthcare continues to grow, integrating adaptive shared control systems in medical robots presents a promising solution for enhancing the accuracy, safety, and reliability of complex procedures. These systems could help mitigate human limitations, such as fatigue, while maintaining the necessary human oversight that is essential in high risk environments like surgery. Through this exploration, we aim to understand how these frameworks can be applied effectively in medical scenarios, building on the foundational work discussed in the previous chapters.

Medical robots are expected to play an increasingly significant role in healthcare, taking on tasks such as helping with surgeries, dispensing medications, disinfecting spaces, and transporting patients. Unlike humans, robots do not become infected when in contact with contagious patients and can tirelessly perform repetitive tasks with high precision, indicating that automation in the health sector is poised to grow, leading to greater demand for medical robots.

In recent years, the integration of shared control systems in the medical domain has gained considerable attention. Shared control involves a collaborative framework in which human operators and robotic systems contribute to the execution of the task, leveraging their respective strengths. This paradigm is particularly advantageous in medical applications, where the precision and reliability of robotic systems complement the experience and decision-making capabilities of human practitioners [103].

One of the primary motivations for adopting shared control in healthcare is the potential to improve the precision and safety of medical procedures. Robotic systems are known for their ability to perform repetitive tasks with high precision, making them ideal candidates for assisting in complex surgeries [104].

Surgical robotics has been used for more than two decades and is advancing more rapidly than other types of medical robotics [105]. The most well-known system is the da Vinci Surgical System [106], which provides surgeons with advanced instruments to perform minimally invasive robotic assisted surgeries. Surgical assistant robots can improve the effectiveness of procedures, allowing surgeons to seamlessly switch between robotic and human control, use precise surgical instruments, and teleoperate while accessing information as surgery progresses [107]. In other words, by sharing control, surgeons can retain decision-making authority while relying on robotic assistance to execute delicate maneuvers, thus reducing the risk of human error and the cognitive load on the surgeon [108],[109],[110].

Suturing is one of the most challenging and time-consuming surgical subtasks [111], made even more difficult when using surgical robots due to limited vision and haptic feedback [112]. Despite these difficulties, the repetitive nature of suturing makes it a good candidate for automation. For example, Liu et al. [113] proposed a systematic method to quantitatively compare different needle grips and entry port placements to autonomously perform the suturing task. Their approach employs a feasibility algorithm to determine the success of a given reference needle trajectory, considering the robot's joint angle and joint torque limits, as well as potential collisions between robot links, tissues, and inner surfaces of the body.

In this area a semi-autonomous system is proposed in [114], where the surgeon specifies the location of the suture and the robot performs the task autonomously while the surgeon supervises and intervenes if needed. In another collaborative robot-human approach [115], a mechanical needle guide is proposed to optimize the needle size, trajectory, and control parameters. Their proposed guide algorithm was evaluated on the da Vinci research kit and resulted in faster suturing performance.

However, in another approach [116] for automating suturing, 3D information from a single clinical endoscope camera is used to track the needle and the set of entry and exit points predefined by a surgeon to perform suturing. Although promising results were reported, the method had limitations due to its high sensitivity to lighting conditions. In another study, instead of asking the surgeon to indicate the insertion points, an automated method was proposed to place these points around a specific incision in an evenly distributed way, and then the Smart Tissue Anastomosis Robot (STAR) placed the sutures on a planar phantom tissue [117].

In the literature, needle grasping is also studied as it is one of the key subtasks of autonomous suturing. For example, in [118] a tracking-by-detection framework is developed to obtain the 3D position of the needle,in which the needle is marked with three green markers; Once the needle is detected, the surgical tool moves closer to the

needle based on visual feedback and grasps the needle through path planning based on observed surgical practice.

The potential applications of shared control in the medical domain extend beyond surgery and suturing. Rehabilitation robotics, for example, benefit from shared control by allowing therapists to guide robotic systems in providing personalized treatment to patients [119]. This collaborative approach ensures that therapeutic interventions are tailored to the specific needs of each patient, leading to improved outcomes.

In short, shared control in the medical domain offers significant advantages by combining the strengths of human expertise and robotic precision. As technology continues to evolve, it is expected that shared control systems will play an increasingly prominent role in enhancing the quality and efficiency of healthcare delivery.

In this chapter a learning from demonstration shared control framework is proposed that can execute suturing movement primitives based on the given insertion points selected by the user through a graphical user interface.

This framework uses only a few demonstrated suturing samples to learn and generate the desired movement trajectories. The movement trajectory generation happens in real-time and is flexible giving different insertion points. This framework is implemented in a Da Vinci surgical tool that is attached to a Franka Emika robotic arm.

5.2 Shared Control for Surgical Needle Insertion

In this section, the implementation and application of the adaptive blended shared control framework outlined in Section 4.2 are examined, specifically focusing on its use in medical suturing. A detailed exploration of the methods and tools used to bring the framework to life is provided, which demonstrates how precision and efficiency can be enhanced in suturing tasks. In addition, the practical implications of utilizing

this framework in a medical context are discussed. By the end of this section, a comprehensive understanding of how the proposed framework is applied in real-world medical settings to improve suturing techniques will be provided.

In this task, a 7-DOF Franka Emika robot arm with an attached 3-DOF da Vinci Endowrist needle driver [120] was used. The objective was to insert a surgical suturing needle into the tissue in the desired position and orientation. The physical setup prepared for this task and the task objective are shown in Figures 18 and 19.

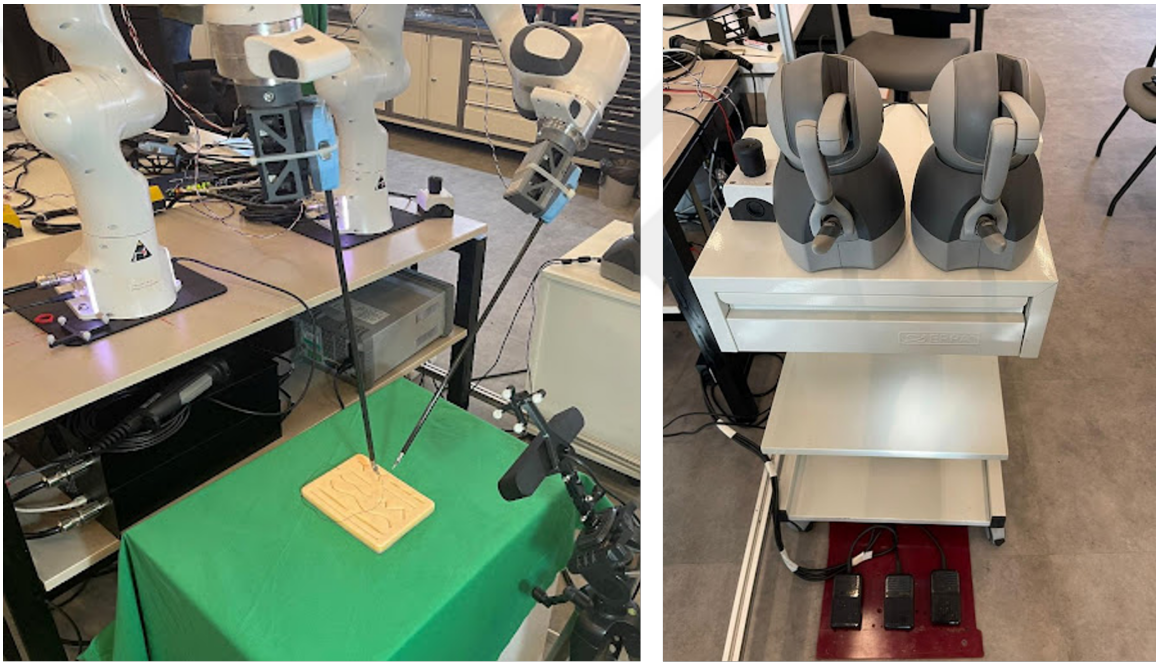


Figure 18: Surgical needle insertion task environment - Teleoperation system and physical setup

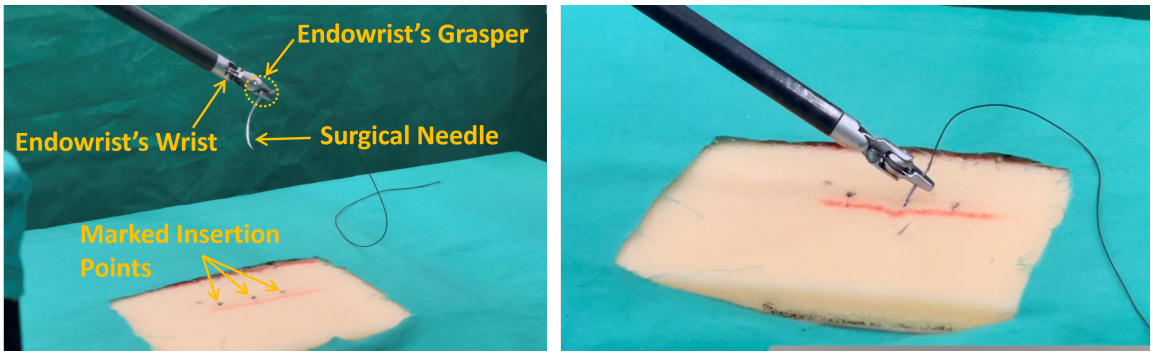


Figure 19: Surgical needle insertion task objective - Initial and final pose

5.2.1 Teleoperation

To teleoperate the robotic arm, a haptic device and two pedals have been used as the interaction tools between the teleoperator and the manipulator. As the haptic device workspace is much smaller than that of the manipulator, the teleoperator might need to move the haptic device freely to achieve the desired end effector pose of the robot. For this purpose, a pedal is added to the system so that the manipulator only receives input from the haptic device as long as the pedal is pushed [121]. Because the picking up and insertion operations require a lot of dexterity on the surgical tool of the system, another pedal is added so that the user can switch between the orientation frames of the robotic system. Therefore, one of the pedals enables the user to control the orientation of the 10-DOF combined system and the other one enables the control of the surgical tool alone where both of them are responsible for clutching, which means that the user can move the haptic device freely to a comfortable position when no pedals are pushed. If data transfer from the haptic device to the robot is active (that is, one of the pedals is pushed), the angular and linear velocities of the haptic device are sent to the 7-DOF serial manipulator velocity controller and the 3DOF surgical tool joint position controller. The serial manipulator velocity controller is implemented with the help of the Franka ROS library [122]. To prevent abrupt velocities and accelerations in the robotic arm, input velocities are filtered before being fed to the manipulator (16).

$$V_{filtered} = \alpha V_{filtered} + (1 - \alpha) V_{input} \tag{16}$$

Here, $0 \leq \alpha \leq 1$ is a scaling constant, $V_{filtered}$ is the resulting filtered velocities, and V_{input} indicates the row velocity readings.

The surgical tool is designed to have a high level dexterity to match the human wrist through a pulley and cable system that creates a coupled kinematic structure.

In particular, the 3-DOF surgical tool is actuated using four Dynamixel XL-320

[123] servomotors. Kinematic decoupling is achieved through a transformation matrix that converts the three desired input angles of the joint to four motor angles so that the desired angles are achieved at the end effector, as described in [124, 125]. The data flow of the teleoperation system is schematized in Figure 20 and the teleoperation setup is shown in Figure 18.

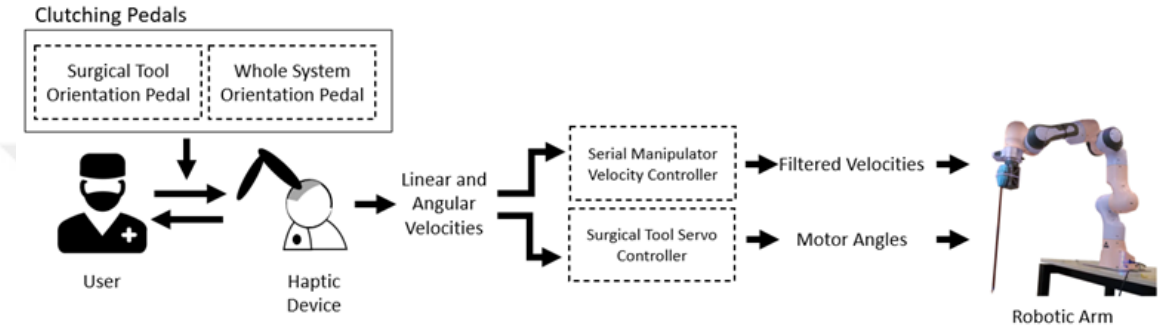


Figure 20: Data flow of the teleoperation system: The user may choose to only control the orientation of the surgical tool or the whole system by pushing to the respective pedals. If the user is pressing one of the pedals the clutching is attained and linear and angular velocities from the haptic device are sent to manipulator's and surgical tool's controllers. Image credit: [124]

5.2.2 Collecting Demonstrations

Trajectory demonstrations were collected as the robot was teleoperated to insert the needle into three desired insertion points marked on the tissue by an expert.

A demonstration trial was started with the user expert pressing the first pedal, which triggered data collection. The user pressed the second pedal to end and save the demonstration. During teleoperation the cartesian information of the wrist (end point of the needle driver poll) and the hand (end point of the gripper), were recorded.

The demonstrations always started at the same initial point and three demonstrations were collected for each insertion point. The frequency of data collection was 30 Hz. The operator then subjectively evaluated the demonstrations after successful execution with a score between 0-100.

The evaluation scores were used to weigh the demonstrations collected for training (see Algorithm 1). The demonstrations were pre-processed to reduce noise and ensure uniform length (in this experiment, the length of trajectories was set to 750). These demonstrations are shown in Figures 18 and 21.

5.2.3 Autonomous Skill Acquisition

Using the Algorithm 1 and the pre-processed demonstrated trajectories, the weighted-CNMP was trained to learn:

- Cartesian trajectory of the robot's wrist (needle driver's first joint)
- Cartesian trajectory of the robot's hand (needle driver's last joint)
- Gripper status of needle driver

After training, robot trajectories for needle insertion could be generated by conditioning the weighted-CNMP network on the desired pose of the end-effector based on a given desired insertion point. To assess the performance and determine the error of this trajectory generation, 10 random needle insertion points within the range of demonstrated data were generated and provided to the network as conditions. The performance and error of this generation are reported in Tables 8 and 9, where the error is defined as the difference between the desired and final position and orientation of the endowrist end-effector. A sample generated wrist position trajectory by the robot is shown in Figure 21. As shown in the figure, the robot can also estimate the trajectory for interpolated insertion points. A video recording of the robot performing this autonomous suturing insertion task is available at the following link: https://youtu.be/90U$_r lJZKI4$.

5.2.4 Graphical User Interface

An adaptive graphical user interface (GUI) is designed to allow easy modification of insertion points, as shown in Figure 22. The interface comprises two panels. The

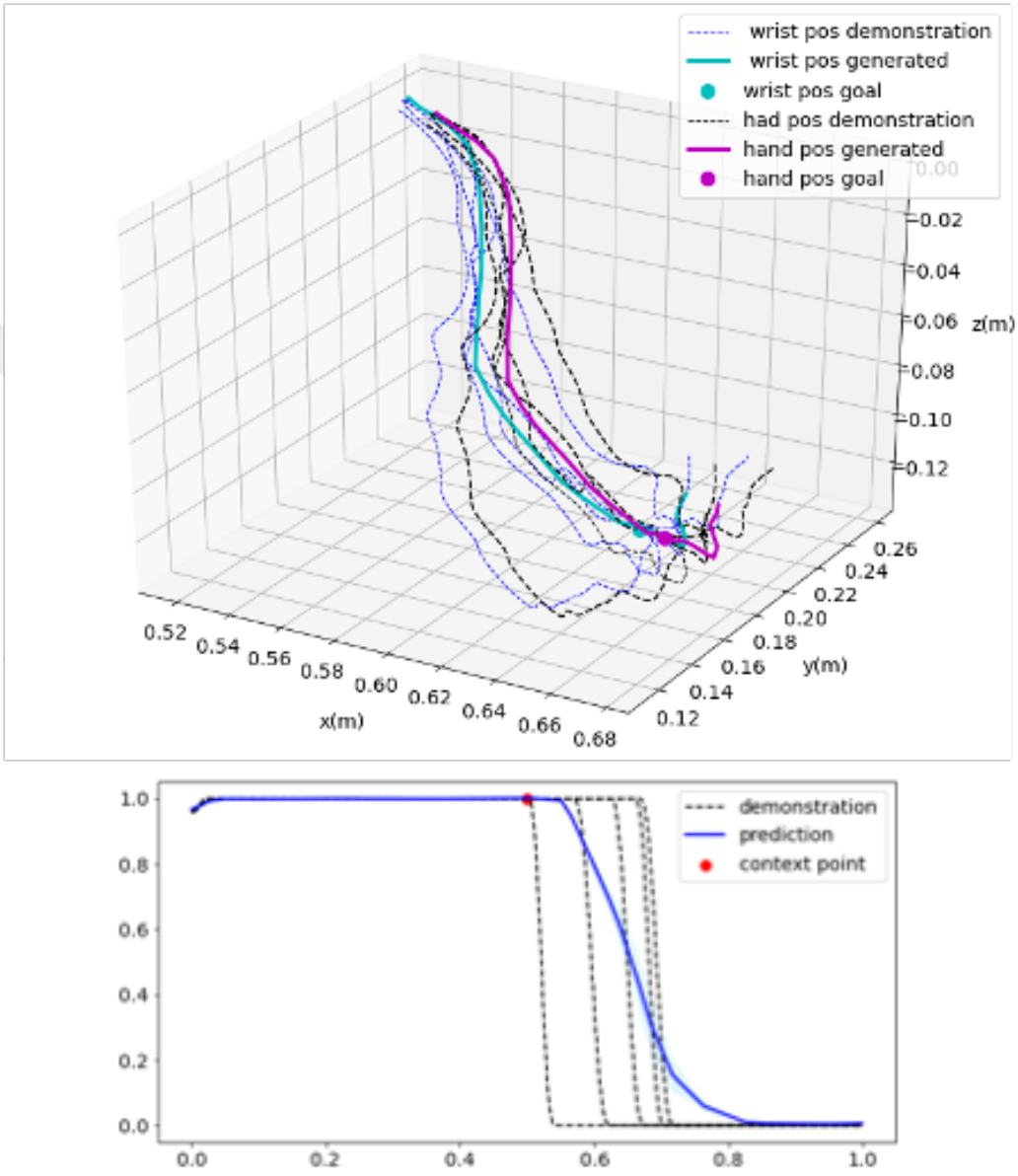


Figure 21: Surgical needle insertion task - Generated trajectories when conditioned on a desired insertion point. Top plot displays the hand and wrist trajectories, and the bottom plot displays the gripper status of needle driver. This figure displays an interpolation case

Table 8: Surgical needle insertion task - Error of autonomous synthesis of wrist trajectory

	Position (mm)		Orientation (rad)	
Axis	Error	stdv	Error	stdv
x	1.55	0.98	0.004	0.003
y	2.38	0.89	0.0017	0.0015
z	3.71	2.01	0.001	0.0008

Table 9: Surgical needle insertion task - Error of autonomous synthesis of hand trajectory

	Position (mm)		Orientation (rad)	
Axis	Error	stdv	Error	stdv
x	0.19	1.57	0.001	0.003
y	1.07	0.55	0.0027	0.0011
z	1.51	1.01	0.001	0.0005

first panel displays an image of artificial tissue marked with grid lines spaced 0.5 cm apart, providing an intuitive guide for editing insertion and exit points in the suture plan. The second panel enables the adjustment of the number of sutures and the positions of the insertion points.

At the GUI startup, the coordinates for the selected number of equally spaced insertion points are listed in the panel. The top left corner of the tissue is set as the origin, and the coordinates are calculated in centimeters. Users can modify the default number of sutures and the coordinates of each individual insertion point. Any change in the number of sutures or insertion point coordinates is visualized on the first panel in real time.

Once the user finalizes the desired insertion points, needle grasp and insertion trajectories are generated by the trained models. If the number of sutures is altered, both the coordinate panel and the visualization panel update according to the new

number. This interface is used to provide different insertion points to the trained models, enabling the generation of trajectories for those points.

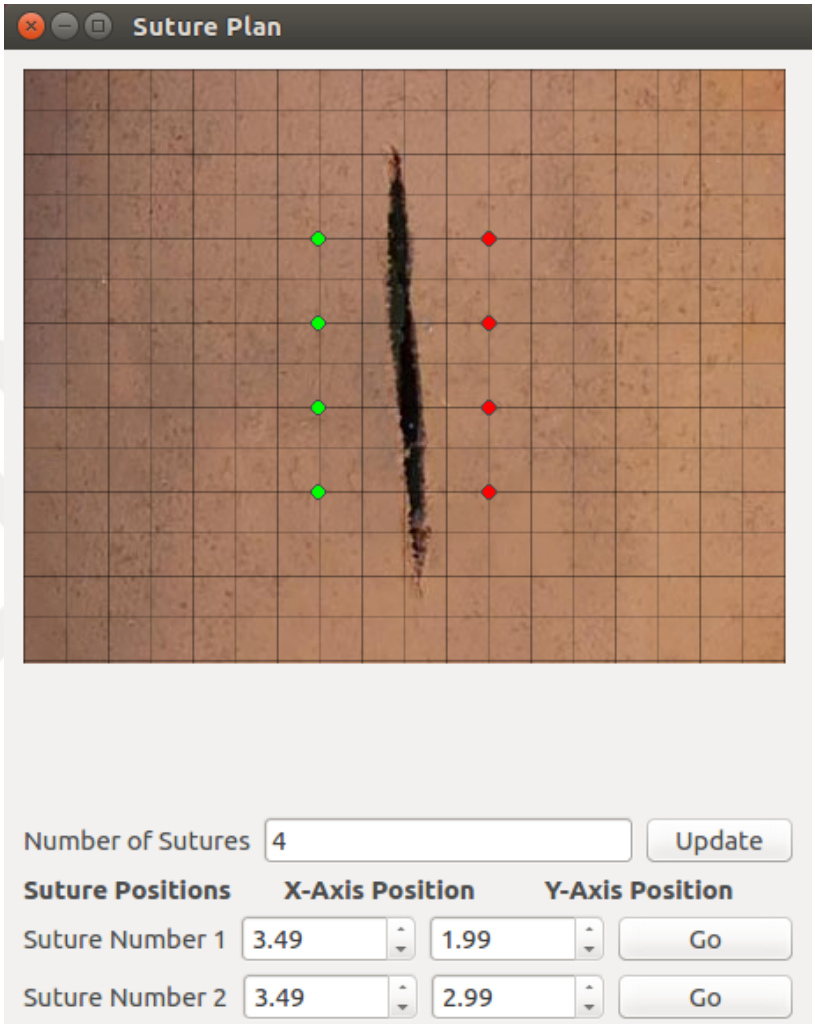


Figure 22: Graphical user interface that allows the user to specify insertion points for trained models to generate trajectory and visualize the positions of the chosen points on the artificial tissue.

5.2.4.1 Learning Scope of Knowledge

In this task, the state of the task was defined as the position of the robot's end-effector. Similarly to the previous task, the states observed in the demonstrations and used during training were labeled as "known." In addition to the 6750 observed states (9 demonstrations for 3 insertion points, each with a length of 750), data

was augmented with 27,000 more states that were randomly generated within the minimum and maximum values for each state parameter and labeled based on their distance to the known states for better generalization. The distance threshold was set to 5 mm. The labeled data were used to test and train the MLP classifier. The data were divided into a training set and a test set, the test set comprising 20% of the data. The classification accuracy on the training data was 0.8830, and on the test data, it was 0.8746. The performance of the MLP classifier is shown in Figure 23.

5.3 Experiment Design

This experiment was designed to evaluate the effectiveness and precision of the proposed adaptive blended shared control framework in performing a surgical needle insertion task, comparing shared control with human-only control. The design details are as follows:

- **Objective:**
  - The primary objective of the task is to accurately insert a surgical needle into tissue at a target point only known by the expert user with the correct orientation. This simulates a common surgical procedure requiring precision and control.
- **Robotic System:**
  - The system includes a 7-degree-of-freedom (DOF) Franka Emika robot arm, which is equipped with a 3-DOF da Vinci endowrist needle driver. This setup allows for detailed control of both position and orientation.
- **Teleoperation Setup:**
  - A teleoperation system, consisting of a haptic device and three pedals, is

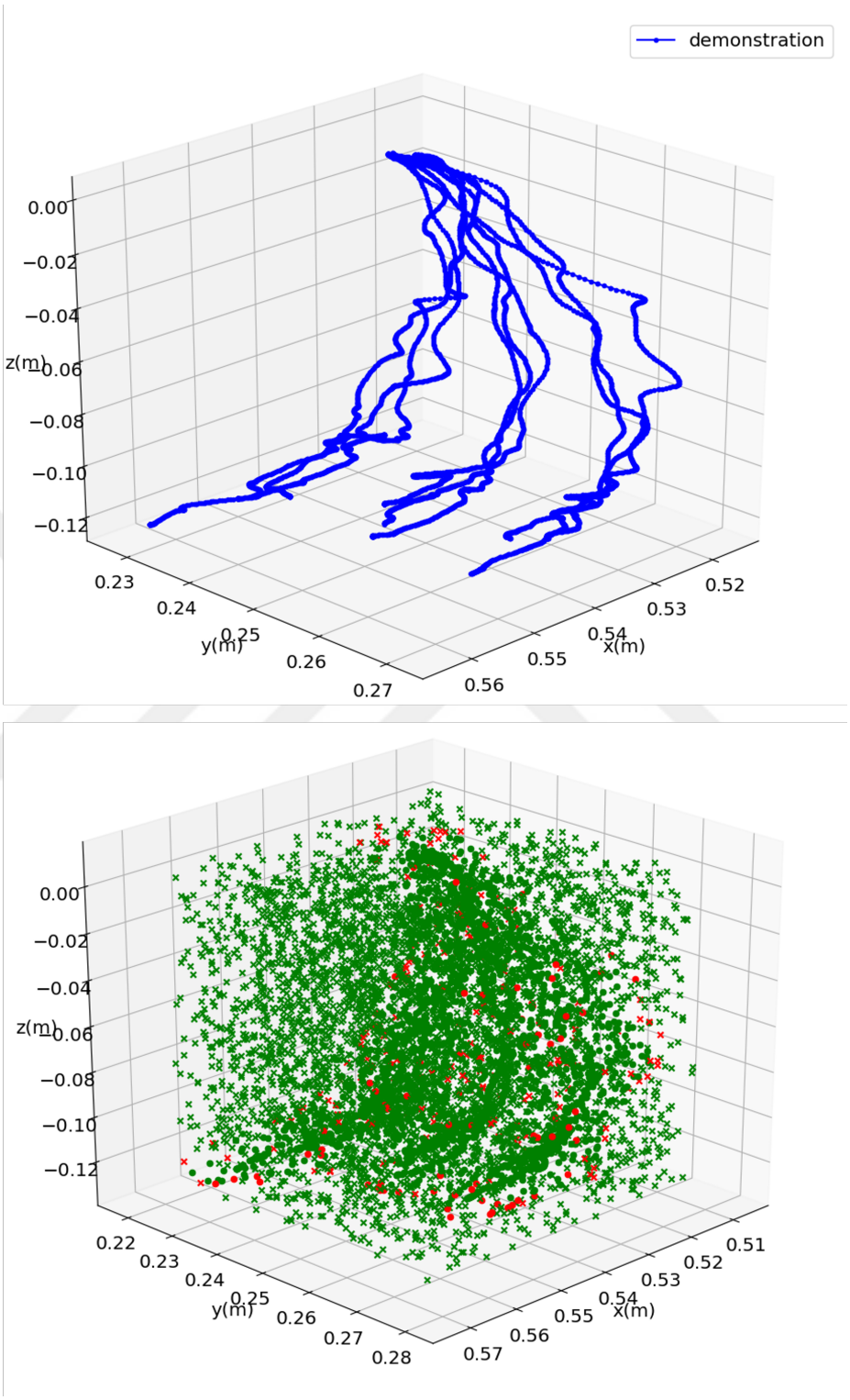


Figure 23: Representation of robot's scope of knowledge -In the top row, the states demonstrated to the robot during training are shown. The row below illustrates the results of state classification, with the symbols "o" and "x" indicating states classified as "known" and "unknown," respectively. States correctly classified are denoted in green, while misclassified states are highlighted in red.

used to control the robot. The haptic device allows for intuitive manipulation of the robot arm, while the pedals enable or disable control over different aspects of the system.

- **Experimental Setups:**
  - **Shared Control:** In this setup, the human operator collaborates with the robot. The system uses an adaptive shared control framework, where the robot estimates the human's intended insertion point and assists in achieving it.
  - **Human Control:** In this setup, the human operator performs the task independently using the teleoperation system, without robotic assistance.
- **Performance Metrics:**
  - The key performance metrics are completion time, distance to the desired insertion point, and the blending parameter, which indicates the extent of human control in the shared control setup.
- **Participants:**
  - The experiments were conducted by two expert human operators due to the complexity and precision required for teleoperation. Considering the complexity of teleoperation and the demanding precision involved in this task, it is impractical to include inexperienced human users in the experiment. Furthermore, our objective is to assess whether the proposed adaptive shared control framework could outperform the expert human operators and/or reduce their workload. Therefore, the exclusive inclusion of expert human operators is imperative.
- **Task Procedure:**

- Each operator performed the task for three different insertion points on the tissue, twice for each setup.
- Demonstrations were collected for training, with trajectories being pre-processed to reduce noise and ensure consistency.

The experiment aims to highlight the potential benefits of adaptive shared control in improving surgical precision and reducing the workload on human operators.

5.4 Results

The results of the surgical needle insertion task are summarized in Table 10. The table presents the mean and standard deviation of key performance metrics for both the shared control and human control setups.

- **Completion Time:**
  - The average completion time for the shared control setup was 30.87 seconds with a standard deviation of 7.411 seconds, whereas the human control setup required an average of 35.101 seconds with a standard deviation of 10.696 seconds.
  - This indicates a notable reduction in task completion time when using the adaptive shared control system, demonstrating its effectiveness in enhancing operational efficiency.
- **Precision (Distance to Target Point):**
  - The shared control setup achieved a mean distance of 4.332 mm from the target insertion point, with a standard deviation of 2.160 mm. In contrast, the human control setup resulted in a mean distance of 6.660 mm, with a standard deviation of 2.334 mm.

Table 10: Descriptive statics of performance metrics in the successful trials in both setups for surgical needle insertion task

	Shared Control		Human Control	
Measures	Mean	std. dev	Mean	std. dev
Time (s)	30.87	7.411	35.101	10.696
Distance (mm)	4.332	2.160	6.660	2.334
Blending Parameter	0.482	0.069	1	0

  - This improvement in precision indicates the advantage of incorporating robotic assistance in this task, as it allows for more accurate needle placement.

- **Blending Parameter:**
  - The blending parameter for the shared control setup averaged at 0.482 with a standard deviation of 0.069, indicating that the human operators contributed approximately 48.2% towards task execution.
  - This metric reflects the balance between human and robotic contributions, emphasizing the collaborative nature of the shared control system and its capability to reduce human workload.

Overall, as shown in Table 10, the adaptive shared control system enhanced both the efficiency and precision of the surgical needle insertion task. The results suggest that integrating robotic assistance with human expertise can substantially improve performance in tasks requiring high levels of precision and coordination.

5.5 Discussion

The results of this experiment point out the advantages of using an adaptive shared control system for the surgical needle insertion task. The reduction in average completion time in the shared control setup demonstrates the system's ability to enhance

operational efficiency. This improvement suggests that robotic assistance can optimize task performance by allowing seamless collaboration between the human operator and the robot.

Precision is another critical aspect that benefits from the shared control system. The improvement in the mean distance from the target insertion point indicates the ability of the system to improve accuracy, which is vital in surgical tasks.

The collaboration between human operators and robots is reflected in the blending parameter, which shows a balanced contribution from both the human and the robot. This balance allows the system to adaptively adjust control levels based on task requirements and operator actions. As a result, the cognitive and physical workload on the human operator is reduced, while the essential element of human judgment remains in complex decision-making processes.

The success of the shared control system in this task suggests its potential applicability to other domains requiring precision and efficiency. Its adaptive nature means that the control framework can be tailored to various tasks, offering improvements over traditional teleoperation methods.

Despite these results, this particular experiment is limited by the small sample size and the exclusive use of expert operators. Future research should include a broader range of participants with varying levels of expertise to assess the system's effectiveness across different skill levels. Additionally, further exploration into optimizing the blending parameter could provide deeper insights into maximizing the advantages of shared control systems.

5.6 Conclusion

This chapter demonstrates the effectiveness of the proposed adaptive shared control system in improving task performance during surgical needle insertion. By integrating human expertise with robotic precision, the system enhances both efficiency and

accuracy, crucial for surgical applications. The shared control approach balances the human operator's input with robotic support, reducing workload and maintaining precision in execution.

A key feature of the system is its ability to continuously estimate the intentions of the human operator throughout the procedure. This real-time estimation allows the robot to dynamically adjust its behavior and role in response to the operator's actions and needs. Depending on its level of confidence in accurately interpreting the operator's intentions, the robot can assume different roles: it may act passively, allowing the human to take full control; take a leading role when confident in its understanding of the task requirements; or engage in a collaborative role, where it shares control with the human to optimize task performance. This adaptability ensures a seamless integration of human and robotic efforts, enhancing the overall effectiveness and efficiency of the operation.

5.7 Future Work Directions

Future research will focus on expanding the applicability and robustness of the adaptive shared control system. Key areas of exploration include:

- **Adaptation for Varying Expertise Levels:** The system will be adapted to accommodate operators with different levels of expertise, broadening its use in training environments and making it more accessible to novice users. This includes designing the system to allow for a gradual learning curve.

- **Guidelines for Human Operators:** Providing human operators with prior knowledge about the autonomous system's behavior during collaboration is considered beneficial. This strategy acknowledges that adapting to a shared control system is a gradual process, as evidenced by our human study where a significant number of failures occurred during initial trials. Guidelines can expedite adaptation, minimizing the learning curve.

- **Application to Other Tasks:** The framework will be explored for application to other complex, precision-oriented tasks beyond surgical needle insertion. This involves tailoring the system to different tools and environmental conditions, extending its versatility.

- **Sensory Feedback Mechanisms:** Implementing advanced sensory feedback mechanisms could enhance responsiveness and accuracy, providing real-time data that informs better decision-making processes.

- **Real-World User Studies:** Conducting user studies in real-world settings will be crucial for evaluating the benefits and limitations of the system in specific scenarios.

- **Efficient Neural Network Models:** Developing more efficient, lightweight neural network models could reduce computational demands and improve real-time processing capabilities, enabling smoother interactions between human operators and the autonomous system.

CHAPTER VI

RESERVOIR COMPUTING FOR LEARNING FROM DEMONSTRATION AND SHARED CONTROL

6.1 Introduction

Throughout the previous chapters, we have explored various aspects of human-robot shared control systems and their applications. Chapter III examined how human adaptation to shared control frameworks enhances task performance and efficiency by integrating human input with robotic support. We demonstrated that adaptive systems, capable of interpreting and responding to human intentions in real-time, improve overall performance and safety.

In Chapter IV, we looked into the specific application of these adaptive systems in surgical needle insertion, highlighting how real-time estimation of human intentions allows robots to dynamically adjust their roles, leading to better precision and reduced operator workload.

Building on these findings, Chapter V introduced the challenges associated with Conditional Neural Processes (CNPs), particularly their limitations in handling extrapolation beyond observed data ranges. While CNPs are effective for fitting observations, they struggle with accurate predictions in novel situations.

This chapter shifts focus to address these limitations by exploring reservoir computing as an alternative approach. Reservoir computing can effectively manage extrapolation issues and adapt quickly to new data, making it a valuable tool for learning movement primitives and enhancing shared control systems.

By incorporating reservoir computing into our frameworks, we aim to build on the adaptive and collaborative principles established in previous chapters, offering a more

flexible and efficient solution for human-robot interaction in varied and challenging tasks. This chapter explores how reservoir computing can overcome the shortcomings of CNPs and contribute to the advancement of intelligent shared control systems.

Empowering robots to learn and execute complex movements in flexible and robust ways requires that methods be developed for efficient movement primitive representations. Learning movement primitives involves the approximation of movement trajectories, which are often described in terms of nonlinear functions [126]. This has been a prominent topic in robotics, with significant efforts being spent on it. Various methods for movement primitive representation utilize the learning from demonstrations paradigm [98], where robots learn to execute movement primitives by imitating an expert who demonstrates the movement.

As explained previously, Conditional neural movement primitives (CNMPs) [2] are built on top of CNPs [1] and are designed as a robotic movement learning and generation method. CNMPs can be conditioned on single or multiple time steps to generate trajectory distributions that satisfy the given conditions. In [125], the model was modified to learn from weighted trajectories and to be used in an adaptive shared control framework [56]. However, these models are still prone to extrapolation and are computationally very expensive.

In this chapter, with the goal of developing a method that is conceptually simple and computationally inexpensive, Echo State Networks (ESNs) [127] are used as the basis of the movement primitive representation method. Echo State Networks are practical and conceptually simple; they employ multiple high-dimensional projections in the large number of states of the reservoir with strong nonlinear mapping capabilities to capture the dynamics of the input. ESNs are random, large, fixed recurrent neural networks (RNNs) [128] that work based on the reservoir computing framework [129].

Recurrent Neural Networks with Parametric Biases (RNNPB) [130] can learn

multiple behavior schemata that are distributively encoded in a single network, where the schema is characterized by the Parametric Biases vector, playing an essential role in the generation and recognition of patterns.

ESNs are a highly practical approach and are computationally much less expensive compared to RNNs training, as the only weights that are modified during training are those that connect the reservoir to the output layer. ESNs have been used in [131] to generate motor trajectories, where the network is combined with a perception module and a controller that can influence its dynamics.

Additionally, in [132], ESNs have been applied to directly control the movements of the antenna of a simulated hexapod, demonstrating that ESNs can store multiple motor patterns as attractors in a single network and generate novel patterns by combining and blending already learned patterns.

The concept of using context in ESNs is introduced to extend the application of single well-defined dynamics/sequence learning to multiple sequence/trajectory learning, where the context alters the dynamics represented by the reservoir.

The applicability of the proposed method for movement primitive representation and generation is also demonstrated in this chapter.

6.2 Echo State Networks

A typical Echo State Network (ESN) is composed of an input layer, a reservoir, and an output layer. The weights of the input layer and the reservoir are kept fixed, being initialized randomly [133]. In this context, $u(t) \in \mathbb{R}^{N_u}$ denotes the input signal, $x(t) \in \mathbb{R}^{N_x}$ represents the state of the reservoir, and $y(t) \in \mathbb{R}^{N_y}$ indicates the output value. Here, $t = 1, \ldots, N$ is the discrete time, and N is the number of data points in the training dataset. The task is defined as learning a model with output $y(t)$ such that $y(t)$ matches $y_{\text{target}}(n)$ by minimizing an error measure $E(y, y_{\text{target}})$. The typical update equations for the state of the reservoir with leaky integration are expressed

in (17) and (18).

$$\tilde{x}(t) = \tanh(W^{\text{in}}[1; u(t)] + Wx(t-1)) \tag{17}$$

$$x(t) = (1-\alpha)x(t-1) + \alpha\tilde{x}(t), \tag{18}$$

where α represents the leaking rate, and $\tilde{x}(t) \in \mathbb{R}^{N_x}$ is the update. The matrices $W^{\text{in}} \in \mathbb{R}^{N_x \times (N_u+1)}$ and $W \in \mathbb{R}^{N_x \times N_x}$ denote the input and reservoir weight matrices, respectively. The output value is typically obtained by equation (19); however, in our implementation, the readout equation has been defined as in equation (20) to ensure that regression is based purely on the reservoir dynamics.

$$y(t) = W^{\text{out}}[1; u(t); x(t)] \tag{19}$$

$$y(t) = W^{\text{out}}[1; x(t)] \tag{20}$$

Here, $W^{\text{out}} \in \mathbb{R}^{N_y \times (N_x+1)}$ (based on equation (20)) is the output weight matrix, which can be computed by linear regression.

6.3 Context-based Echo State Networks

To extend the application of ESNs from single sequence learning to multiple sequence learning, the insertion of context in the input signal is proposed to alter the reservoir state and estimate the output accordingly. This approach is termed as Context based Echo State Networks (CESNs)[134].

For generating movement primitives using CESNs, the network is provided with multiple movement trajectories coupled with a context. Conceptually, context can describe the objective or a feature of a movement. For example, the context could define the target position that the movement should reach or the position of an obstacle that the movement should avoid.

In online robotic applications, robust execution of movement primitives and resistance against external disturbances can be enhanced through real-time feedback. Hence, the integration of context and feedback forms the input signal to CESN. That being said, use of feedback is for the benefit in robotics application but the network can perfectly work without feedback (i.e. with open loop execution). The architecture of CESN is illustrated in Figure 24.

Suppose that $c(t) \in \mathbb{R}^{N_c}$ denotes the context and $f(t) \in \mathbb{R}^{N_f}$ denotes the feedback, then the input signal $u(t)$ is formed as in (21).

$$u(t) = [c(t), f(t)] \tag{21}$$

The training dataset contains multiple trajectories with their corresponding contexts. Depending on the type and use of the context, it can stay constant or change over time for each trajectory in the dataset.

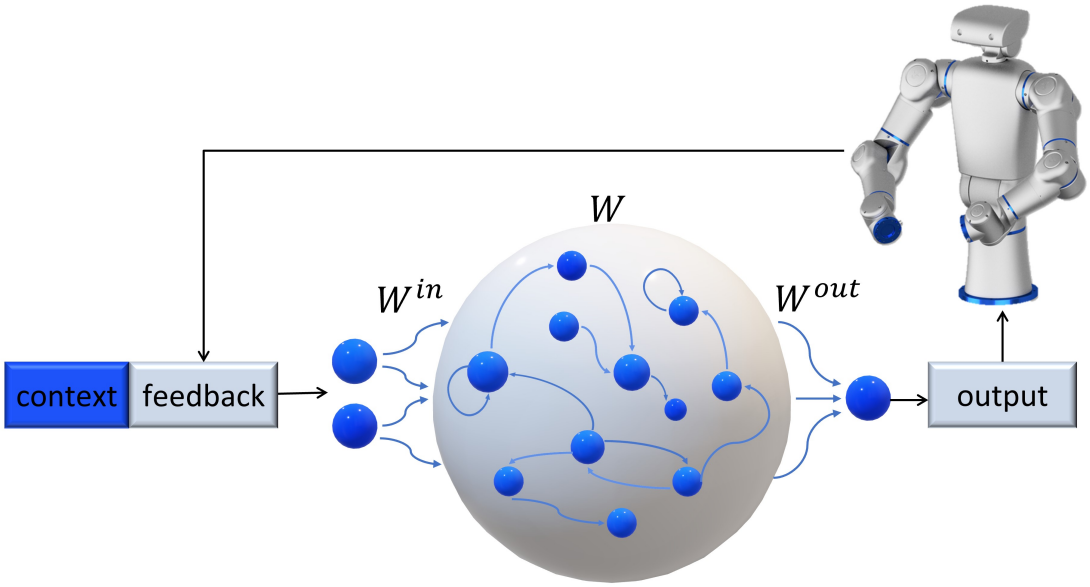


Figure 24: Context based Echo State Network architecture

6.3.1 Data Augmentation with Noise

Feedback received from the system during online task execution can often be noisy or delayed, making the acquisition of perfect feedback infeasible. To enhance the robustness of the system, the feedback component of the input signal is augmented with noise, as illustrated in equation (22).

$$\begin{aligned} u(t) &= [c(t), f(t)] \rightarrow y(t) \\ \acute{u}(t) &= [c(t), f(t) + \epsilon] \rightarrow y(t) \end{aligned} \tag{22}$$

Here, $u(t)$ and $y(t)$ denote the original input and output, while $\acute{u}(t)$ represents the augmented input with noise ϵ, which should lead to the estimation of the same output $y(t)$. The noise ϵ is randomly generated, following the uniform distribution defined in equation (23).

$$\epsilon = \mathcal{U}[b_l, b_h] \tag{23}$$

In this context, $\mathcal{U}$ represents a uniform distribution with a lower bound b_l and an upper bound b_h. It is demonstrated that this augmentation results in a higher accuracy estimation of movement trajectories.

6.3.2 CESN Implementation

For the experiments conducted in this chapter, the parameters of the Context-based Echo State Network (CESN) were configured as follows:

- The number of reservoir units (N_x) was set to 500. For more challenging tasks, a larger reservoir can be implemented, with reservoir sizes of the order of 10^4 not being uncommon [133], [135].
- Input weights (W^{in}) were randomly generated using a uniform distribution and were scaled based on the range of the input signal [133].

- Reservoir weights (W) were also randomly generated with a uniform distribution and were scaled to ensure the echo state property [127].
- The leaking rate (α) was set to match the time interval between two consecutive time steps in the discrete realization of the task execution [133].

6.4 Experiments and Results

6.4.1 2D Reaching with No Feedback

As an initial step to assess the concept of context in ESNs, a reaching task was designed using the Franka Emika robotic arm [136], with an attachment in the Coppeliasim simulation environment [100]. The objective of this task is to move the robot's end-effector to a given desired target position.

6.4.1.1 Experiment Design

- **Context:** In this task, the context was defined as the target position ($\in \mathbb{R}^2$).
- **Output:** The CESN output was defined as the desired target position ($\in \mathbb{R}^2$) trajectories to reach the corresponding context.
- **Demonstrations:** The training dataset was collected by a human user controlling the robot's end-effector via a computer mouse. The mouse displacement in the x and y axes was mapped to the end-effector position with a scalar (the z-axis remained constant in this experiment), and the movement trajectories to reach 8 different target points on a grid of $0.5 \times 0.5\,\text{m}$ were demonstrated.

The task environment, demonstrated movement trajectories, and the network-estimated movement trajectories are illustrated in Figure 25.

6.4.1.2 Results

The RMSE of training for estimating the end-effector position trajectory was $0.0011\,\text{m}$. The mean end-effector final position distance to different target points, including

known (used in training), interpolated, and extrapolated points, is reported in Table 11.

Table 11: Accuracy of the 2D reaching task using the Franka Emika robot for different target points given as context to CESN.

	Distance to target (m)	
Target	Mean	std. dev
Known (8 points)	0.0011	0.0002
Interpolated (8 points)	0.0014	0.0006
Extrapolated (8 points)	0.0268	0.0203

6.4.2 3D Reaching Task

The objective of this task is to move the Torobo humanoid (Tokyo Robotics Inc) robot's right arm end-effector to a given desired target position in 3D space (see Figure 28). This task was designed to assess the ability of CESNs to generalize and to test the effect of data augmentation with noise on the accuracy of the network estimation.

6.4.2.1 Experiment Design

- **Context:** The context was defined as the desired target position $\in \mathbb{R}^3$ to reach in a 3D grid.
- **Output:** The output was defined as the desired end-effector position trajectory $\in \mathbb{R}^3$ to reach the target point.
- **Feedback:** The feedback to CESN was the current end-effector position $\in \mathbb{R}^3$.
- **Demonstrations:** Synthetic demonstrations in the Gazebo simulation environment [137] to reach 27 points on a 3D grid of $0.4 \times 0.4 \times 0.4\,\text{m}$ were used as demonstrations to train the CESN.

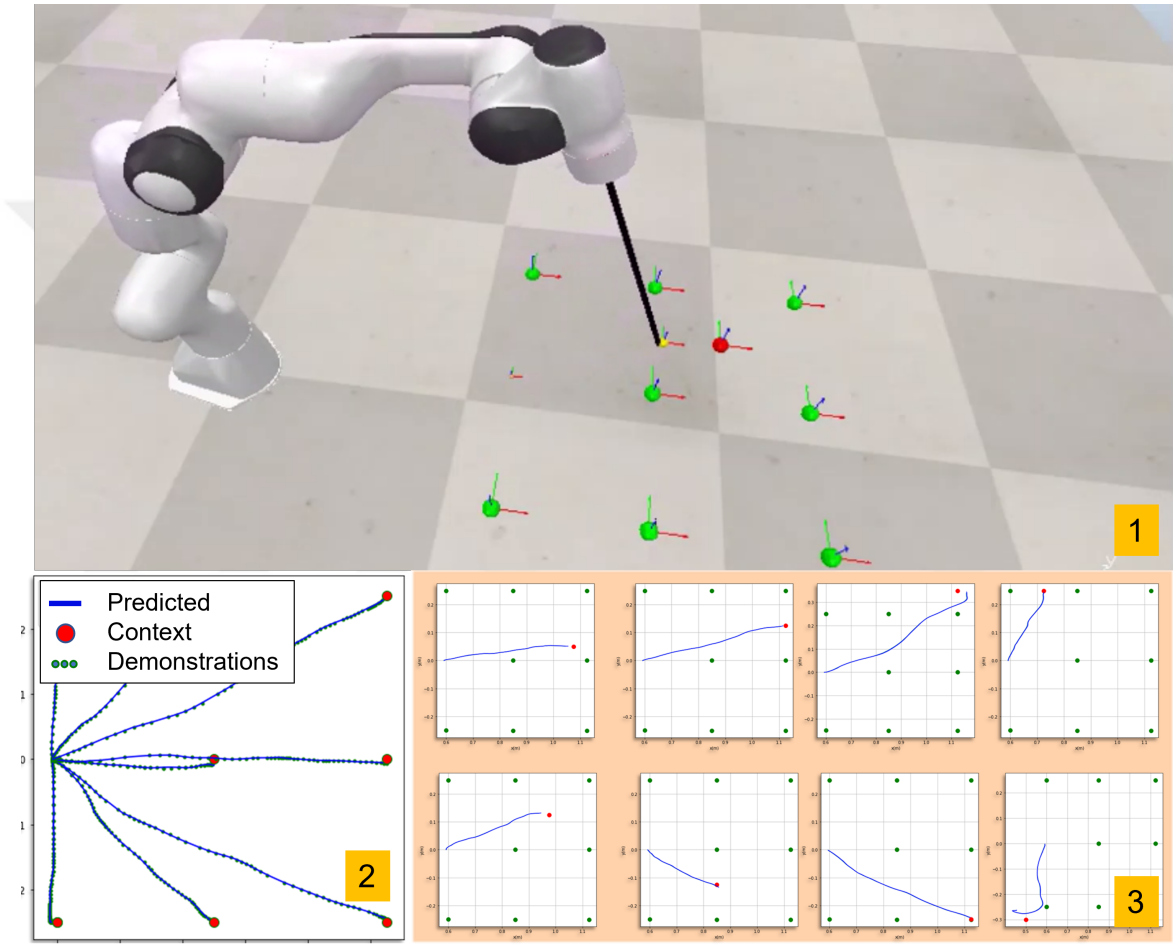


Figure 25: 1- The environment of the reaching task using the Franka Emika robot in Coppeliasim. 2- Demonstrated trajectories in training vs the predicted ones with the given context that were included in the training dataset. 3- Generalization performance; the predicted trajectories for the context not included in the training dataset. 2,3- Plots display the top view of the grid that is 0.5 by 0.5(m).

- **Data Augmentation:**
  - To achieve robustness, the system was tested with the original dataset versus when the data were augmented 5, 10, and 20 times with noise (marked 1x, 5x, 10x, and 20x, respectively).
  - The noise had an initial bound of $[-0.05\,\text{m}, 0.05\,\text{m}]$, which was linearly tapered to 0.0 m as the trajectory progressed over time.

6.4.2.2 Results

The effect of this data augmentation with noise is reported in Figure 26. The training error (RMSE) and the accuracy of the generated movement during testing are compared when using the original dataset versus when the data are augmented. The accuracy of the movement is also compared when the feedback is based on the previous CESN's prediction (predicted feedback) versus when the feedback is the robot's status in real time (real feedback).

Based on the results given in Figure 26, the demonstrated data was augmented 10 times with noise, where the noise was randomly generated with a uniform distribution with an initial lower bound of -0.05 m and an upper bound of $+0.05$ m, and the upper and lower bounds of the uniform distribution for added noise were linearly tapered to 0.0 m as the trajectory progressed over time (see Figure 27). The added noise to the feedback and the CESN estimation of the training data is shown in Figure 27. The RMSE of training was 0.0019 m to estimate the 3D position trajectory of the end-effector.

After training the CESN with augmented demonstrations, the test was carried out on the real robot with different target points (context), where the robot was operated by sending desired end-effector position commands obtained from the CESN output. When testing the performance of the CESN on the real robot, the robot first obtained

Table 12: Accuracy of the 3D reaching task using the Torobo humanoid robot for different target points given as context to CESN.

	Distance to target (m)	
Target	Mean	std. dev
Known (27 points)	0.0099	0.0055
Interpolated (16 points)	0.0096	0.0038
Extrapolated (16 points)	0.0354	0.0211

the 3D position of the desired target by locating a red-colored object using its head-mounted RGBD camera. Subsequently, the obtained position of the object was used as a context, which was coupled with the real-time feedback from the robot at each time step to form the input signal for the CESN generating the next desired trajectory point to reach the red-colored object.

The mean distance to the target position for the known contexts (27 target points) was $0.0099 \pm 0.0055\,\text{m}$, for 16 interpolated contexts $0.0096 \pm 0.0038\,\text{m}$, and for 16 extrapolated contexts $0.0354 \pm 0.0211\,\text{m}$. The result and the robot performance for reaching different target points, including known, interpolation, and extrapolation contexts, are reported in Table 12 and illustrated in Figure 29.

6.4.3 Obstacle Avoidance Task

The objective of the obstacle avoidance task is to move the Torobo humanoid robot's right arm end-effector over a given obstacle with a motion demonstrated by a human, as shown in Figure 30. Cube blocks were used as obstacles in this task.

This task was designed to assess the CESN's ability to estimate more complex and higher-dimensional output using very few demonstrations.

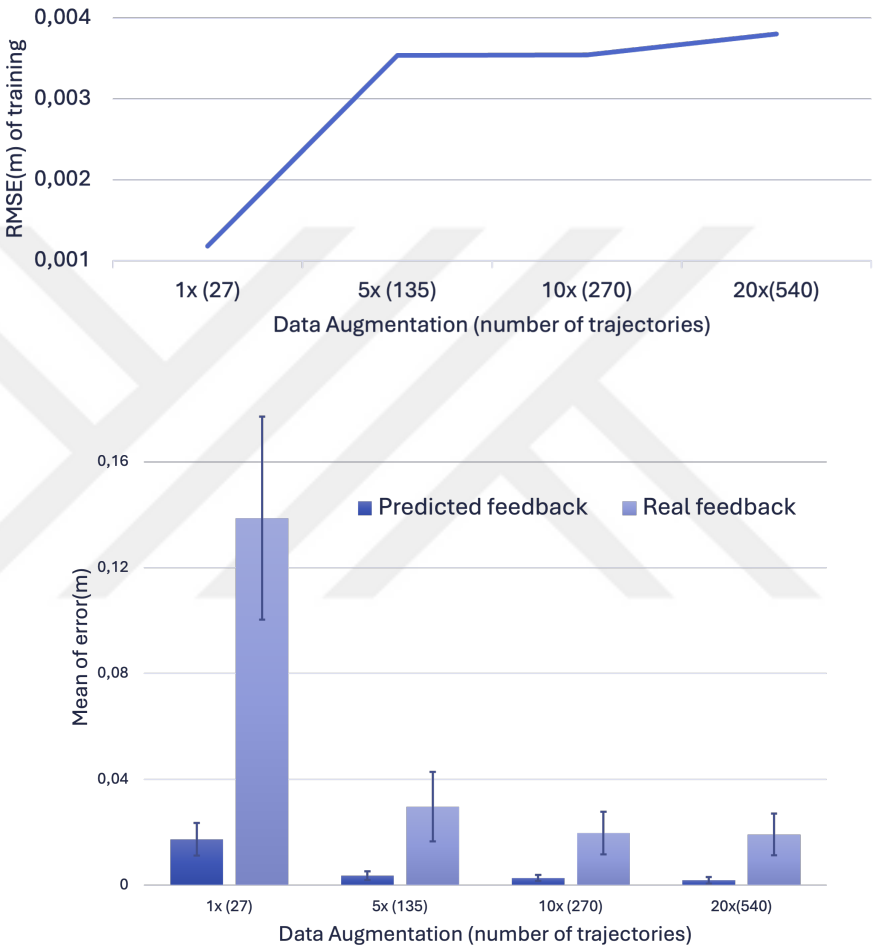


Figure 26: Effect of data augmentation with noise. The top plot reports the RMSE of training, and the bottom plot displays the mean distance (m) to the target position in two cases: using the network prediction as feedback (blue) and using the real-time feedback from the robot (red). Error bars represent standard deviation.

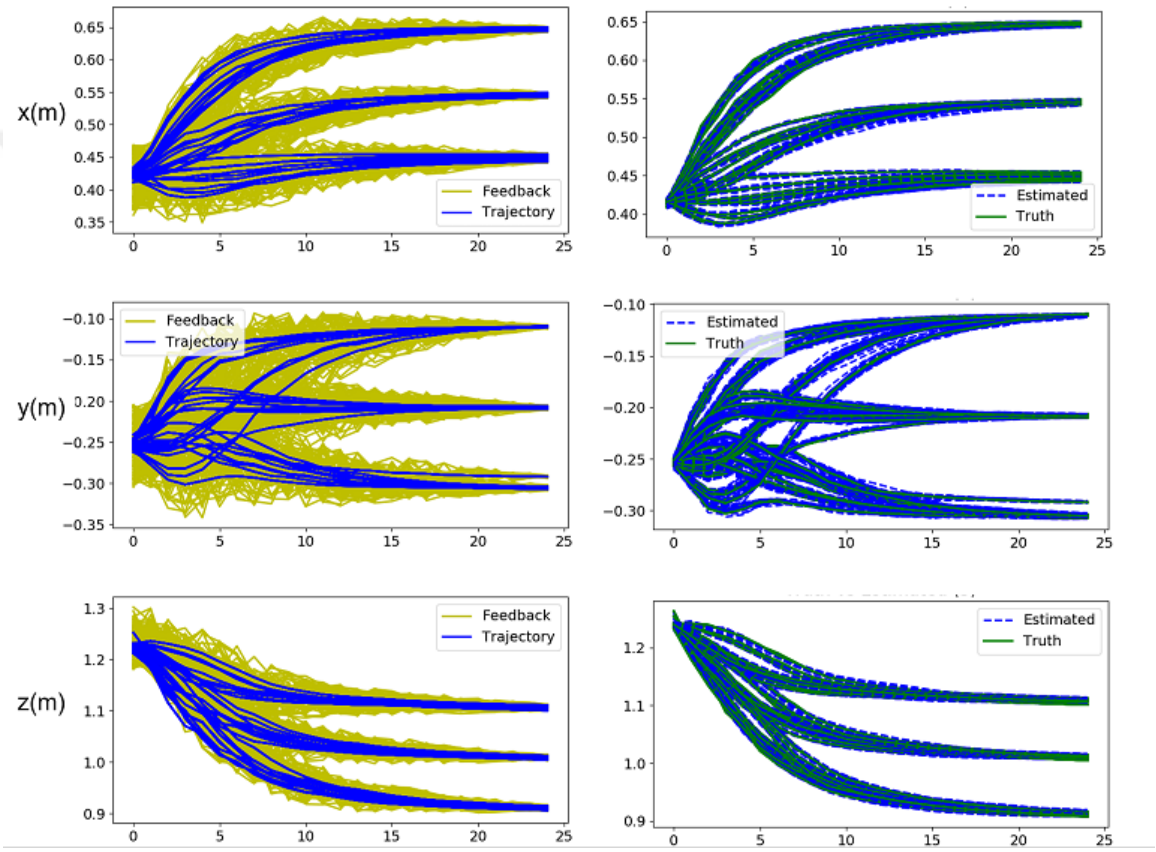


Figure 27: The left side plots display the demonstrated position trajectories in the x, y, and z axes and the augmented feedback with noise. The right side plots display the network estimation and the ground truth of the position trajectories for the reaching task using the Torobo humanoid robot.

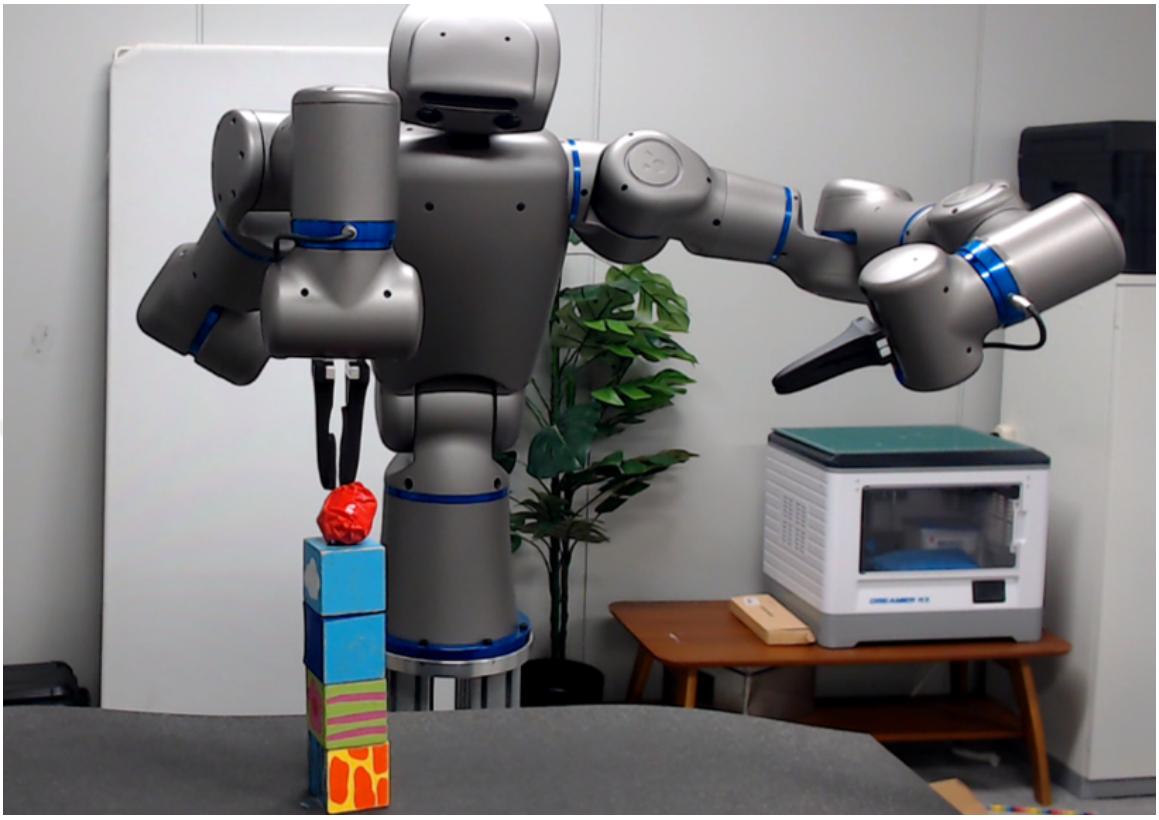


Figure 28: The environment of the reaching task using the Torobo humanoid robot. The robot finds the red object by using its vision system, then the found position of the red object is used as the context.

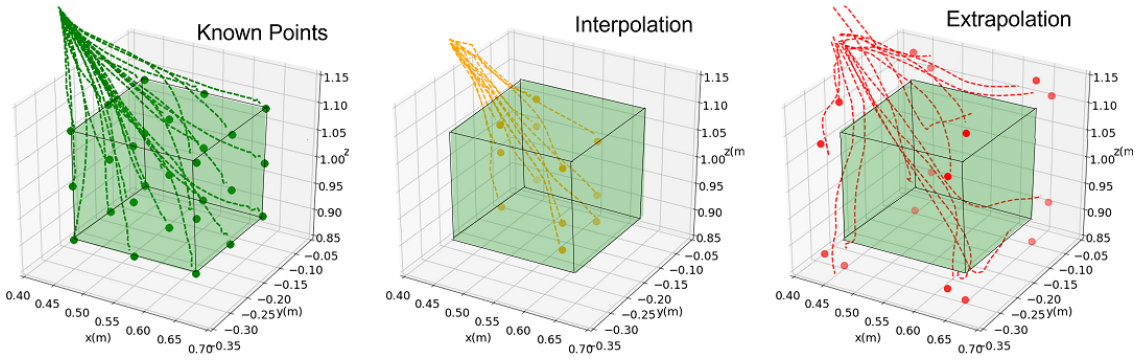


Figure 29: Robot performance in reaching different target points. The green cube displays the region of the demonstrated target points. The green, yellow, and red dots indicate known, interpolation, and extrapolation points, respectively.

6.4.3.1 Experiment Design

- **Context:** The context for CESN was defined as the position of the highest point of the obstacle $\in \mathbb{R}^3$.
- **Output:** The output was the desired joint trajectory $\in \mathbb{R}^{10}$ to avoid the obstacle by passing on top of it.
- **Feedback:** The feedback was the current joint positions of the robot $\in \mathbb{R}^{10}$.
- **Demonstrations:** The movement demonstrations for this task were collected while a human user teleoperated the robot via a haptic touch device, starting from an initial point and passing above the obstacle. Only two movement trajectories were demonstrated by a human (one for 3-block and one for 5-block obstacles) teleoperating the robot.)
- **Data Augmentation:** The demonstrated trajectories were augmented with noise 10 times and used as training data for CESN.

6.4.3.2 Results

The RMSE of training was 0.0148 rad for estimating the robot joint trajectories. After training the CESN, the robot was operated by sending desired joint positions obtained from the CESN output, and the test was conducted by putting different numbers of cube blocks on top of each other, as shown in Figure 30, and giving the top cube position as context to CESN.

The performance of the robot in these obstacle avoidance tasks for known, interpolation, and extrapolation cases is illustrated in Figure 31. As observed in Figure 31, given only two demonstrations for two obstacles, CESNs could generalize very well and generate trajectories for the robot to avoid obstacles almost twice as the maximum height of an obstacle known in the training dataset successfully.

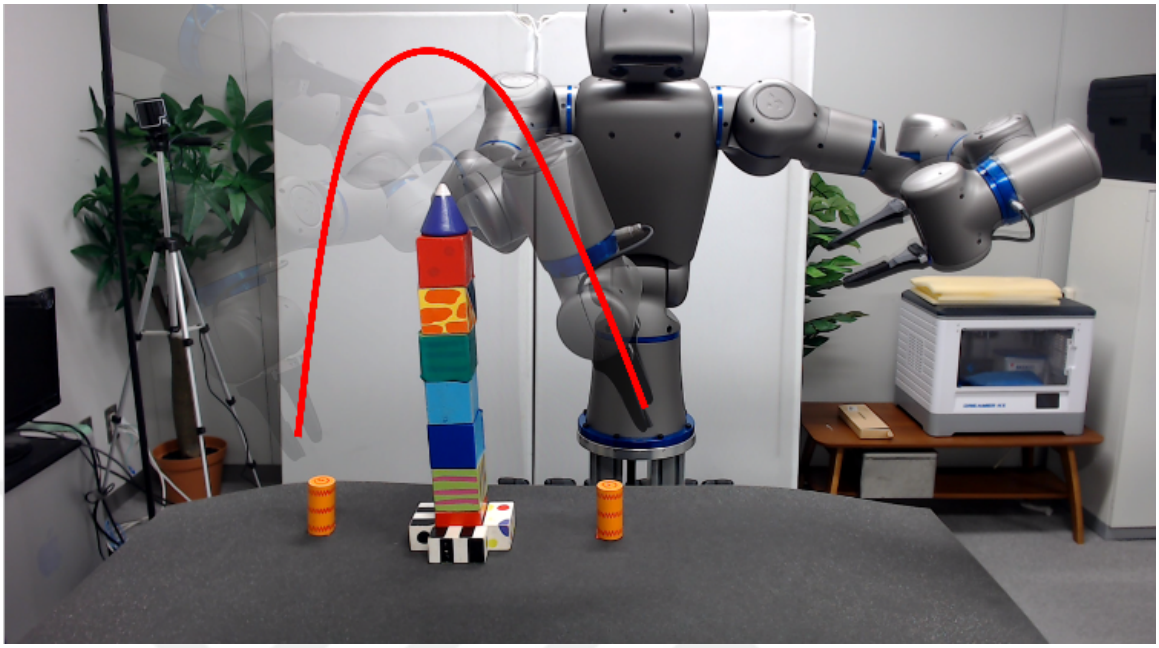


Figure 30: The environment of the obstacle avoidance task using the Torobo humanoid robot. The robot's right arm movement trajectory to avoid 8 blocks as the obstacle is shown with a red line.

6.4.4 Discussion

The 2D reaching task in simulation served as a preliminary exploration of CESN's context-awareness capabilities. Despite the absence of feedback during simulation, CESNs maintained a low RMSE, showing precise trajectory estimation. While the error increased slightly for interpolated and extrapolated target contexts, the CESN demonstrated a reasonable ability to generalize to new points, indicating its potential in applications requiring adaptability. This task highlights the network's strength in handling context variations within a structured environment.

In the 3D reaching task, data augmentation with noise proved to be a significant factor in improving CESN accuracy. By augmenting the training data multiple times with controlled noise, the CESN was able to generalize more effectively across a variety of target contexts, demonstrating a low root mean square error (RMSE) in trajectory estimation. The comparison between using predicted feedback versus real-time feedback highlighted the importance of accurate real-time data in enhancing task

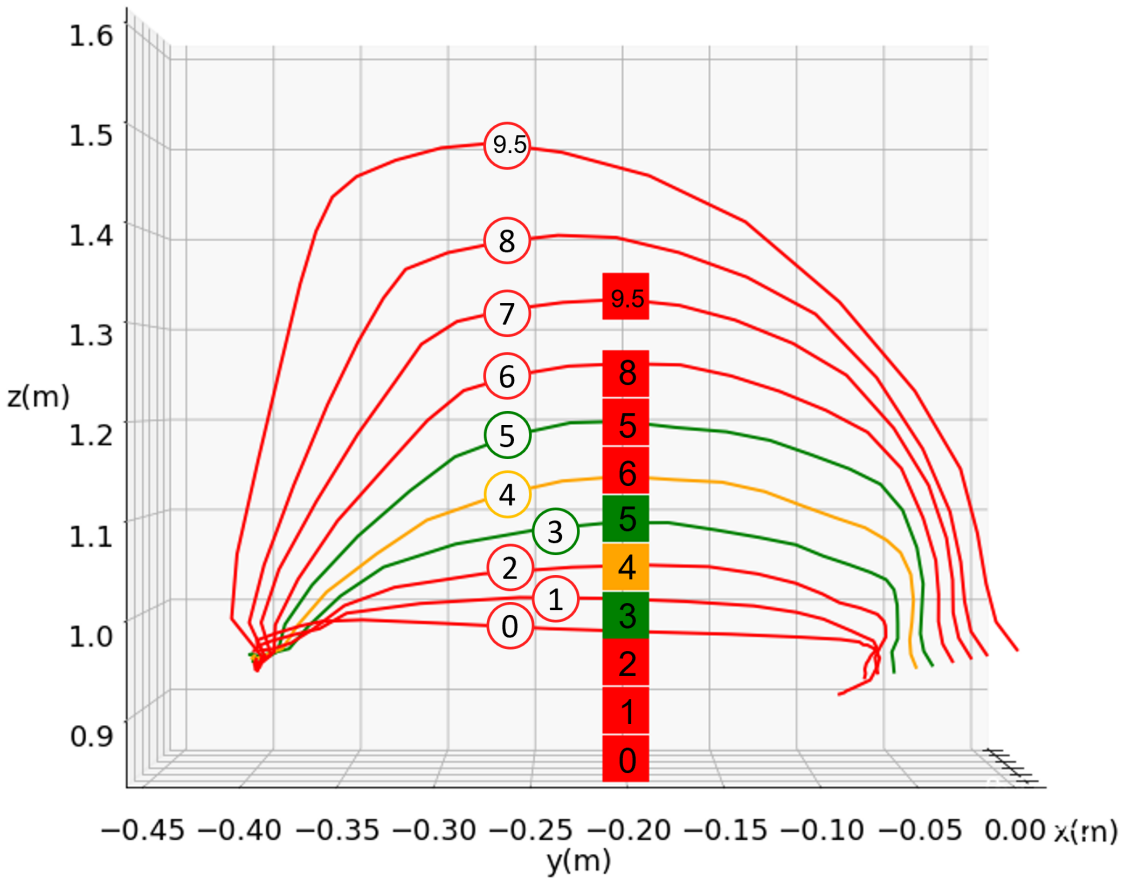


Figure 31: Robot performance in the obstacle avoidance task. Colored squares display the position of cube blocks as obstacles. The number on the squares indicates the number of blocks put in front of the robot to avoid. The corresponding trajectory is marked with the same number. Green, yellow, and red colors show known, interpolation, and extrapolation cases, respectively.

accuracy. This task showcases CESN's ability to learn from limited data, achieving robust results in dynamic tasks.

The obstacle avoidance task presented a more complex challenge due to the higher-dimensional output required for joint trajectory planning. Despite having only two demonstrations augmented with noise, CESNs successfully generated effective trajectories for avoiding obstacles of varying heights. This task demonstrated CESN's efficiency in learning complex motions from minimal data and its potential for extrapolating to novel scenarios that are significantly different from the training set. The ability to generalize from limited data suggests the potential of CESNs for real-world applications, especially where rapid adaptation is necessary.

Overall, the tasks collectively highlight CESN's promise in robotic control, particularly in scenarios requiring efficient learning from limited data and adaptability to new contexts. The results emphasize the importance of data augmentation and the integration of real-time feedback to maximize CESN's performance. However, the limitations observed in extrapolating to unfamiliar situations suggest that further enhancements are needed to improve CESN's robustness in complex and less structured environments.

6.5 CESNs with Prediction Confidence for Human-Robot Shared Control

Estimating prediction confidence in machine learning and robotics is crucial for ensuring the reliability and safety of autonomous systems. In complex real-world scenarios, various uncertainties are present, from noisy sensor data to inherent ambiguity in human interactions.

By quantifying prediction confidence, systems can make more informed decisions, particularly in high-uncertainty situations. For example, in self-driving cars, accurately estimating object detection confidence can help prevent accidents by indicating when the system is uncertain about its surroundings, prompting a cautious response

or the need for human intervention. Furthermore, in robotics applications such as medical diagnosis or industrial automation, understanding prediction confidence enables robust decision-making, thereby reducing the risk of costly errors or adverse outcomes.

The importance of prediction confidence in autonomous systems is heightened in a human-robot shared control context, where human operators and autonomous agents collaborate in decision-making processes.

When control is shared between humans and robots, the human operator relies on the autonomous system to provide accurate predictions and guidance. However, erroneous or overly confident predictions can lead to misunderstandings, conflicts, or even accidents.

Reservoir computing fundamentally involves a fixed, randomly generated recurrent neural network, referred to as the reservoir, paired with a trainable readout layer. Unlike traditional neural networks where all parameters undergo training, in reservoir computing, only the readout layer is optimized while the reservoir's internal connections remain unchanged. This approach simplifies training and accelerates learning.

6.5.1 Prediction Interval

In linear regression, a prediction interval provides a range within which future observations are likely to fall with a specified level of confidence. Unlike confidence intervals, which estimate the mean value of the response variable for a given predictor value, prediction intervals account for both the uncertainty in estimating the mean response and the variability of individual observations around that mean.

The prediction interval considers the uncertainty in the estimated regression coefficients and the variability of the response variable around the regression line. This

involves using the standard error of the regression estimate and the residual standard error, which captures the scatter of data points around the regression line. The calculation is as follows:

$$\hat{Y}_{\text{pred}} \pm t_{\alpha/2} \times s \times \sqrt{1 + \frac{1}{n} + \frac{(X_{\text{pred}} - \bar{X})^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2}} \tag{24}$$

In this formula:

- $\hat{Y}_{\text{pred}}$ is the predicted value of the response variable.
- $t_{\alpha/2}$ is the critical value from the t distribution with degrees of freedom $n - 2$ and a significance level of $\alpha/2$.
- s represents the residual standard error, indicating the scatter of data points around the regression line.
- X_{pred} is the predictor value for which the prediction is made.
- $\bar{X}$ is the mean of all predictor values.
- X_i are the observed predictor values in the dataset.
- n is the number of observations in the dataset.

For multidimensional predictor values, the prediction interval formula in linear regression is modified as follows:

$$\hat{Y}_{\text{pred}} \pm t_{\alpha/2} \times s \times \sqrt{1 + \mathbf{X}_{\text{pred}}^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}_{\text{pred}}} \tag{25}$$

Where:

- $\mathbf{X}_{\text{pred}}$ is the vector of predictor values for which the prediction is made.
- $\mathbf{X}$ is the matrix of all predictor values in the dataset.
- $t_{\alpha/2}$, s, and $\hat{Y}_{\text{pred}}$ have the same meanings as in the previous formula.

The only trainable component of the CESN is the linear readout layer. This layer linearly maps the reservoir states to the desired output, which means training the

CESN reduces to solving a linear regression problem. This linear nature of CESNs allows for the application of prediction intervals, enabling the specification of confidence levels in the output predictions.

The ability to measure prediction confidence makes CESNs well-suited for the proposed adaptive human-robot shared control systems, where prediction confidence serves as the foundation for determining how control is shared between the human operator and the robot.

6.6 Emulating CNMPs with CESNs

To evaluate and compare the performance of Conditional Neural Movement Primitives (CNMPs) and Context-based Echo State Networks (CESNs), four simple trajectories were demonstrated to both neural network models. In the CNMP architecture, the observation layers have a size of [128, 128, 128], and the decoder layers are sized [128, 128, 2]. For the CESN, a reservoir size of 500 was utilized. While CNMPs require a significant amount of time for training, CESNs offer the advantage of nearly instantaneous training. The resulting trajectories generated by these two networks are illustrated in Figure 32. The figure clearly demonstrates that CESNs exhibit superior performance in generating extrapolated trajectories compared to CNMPs, highlighting their effectiveness in handling scenarios beyond the demonstrated cases.

6.6.1 Single context point

To clarify, CNMPs can be conditioned using any desired number of context points, as explained earlier. However, for CESNs, the desired number of context points must be specified before training. In this test, the number of context points was set to 1, and the estimated trajectories and confidence factors of the estimations in both models were compared. The confidence of the estimation in CNMPs is represented by the standard deviation of the estimated normal distribution, with the mean as the given context point. In contrast, the confidence of the estimation in CESNs is represented

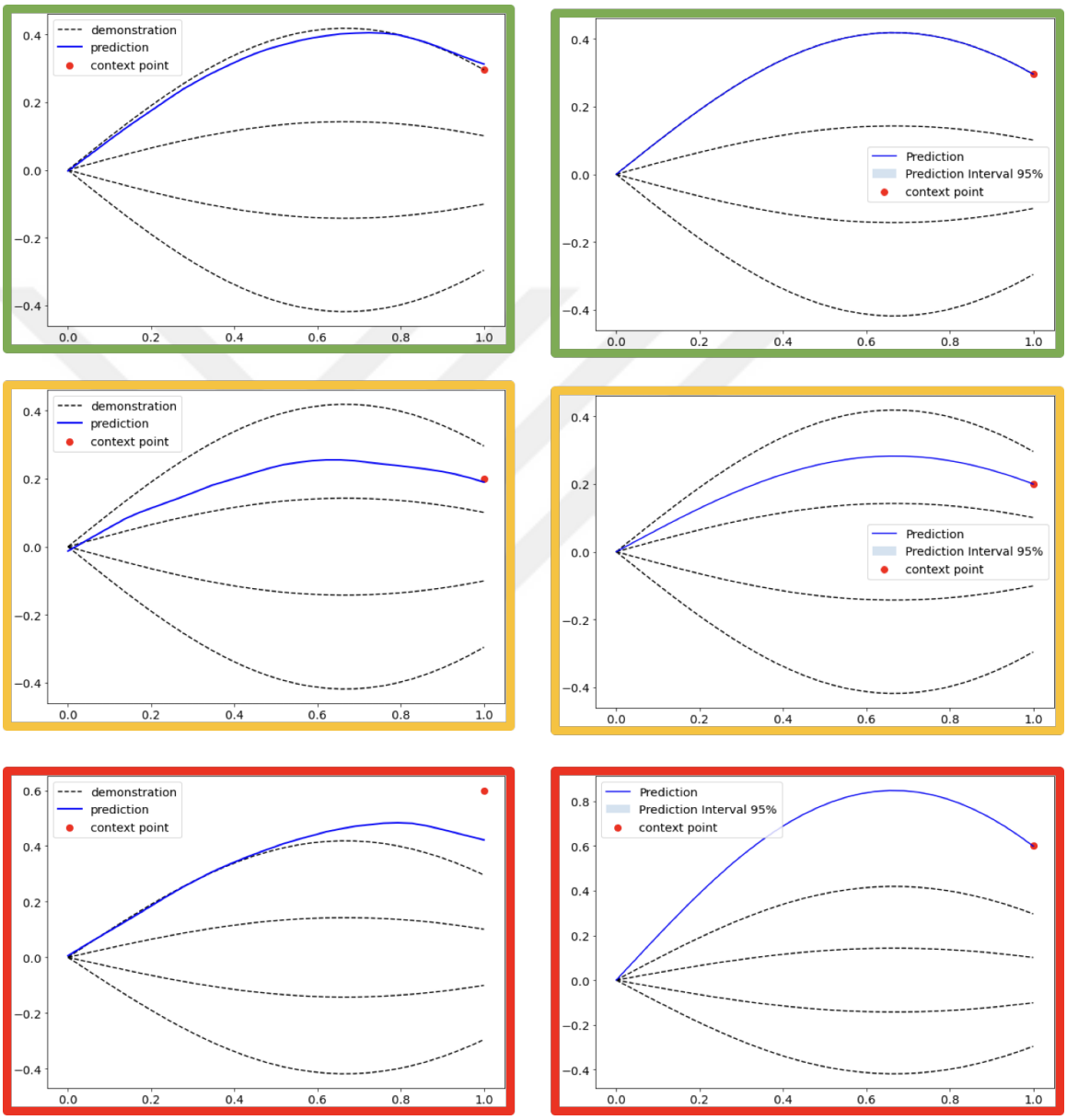


Figure 32: CNMP vs. CESN Predictions: The left plots display CNMP predictions, while the right plots illustrate CESN predictions for scenarios involving known (green), interpolation (yellow), and extrapolation (red) cases, each with one context point.

using a prediction interval. It should be noted that the absolute values of these two factors may not always be directly comparable. However, they can provide a relative indication with respect to their input range, of the confidence in the estimations.

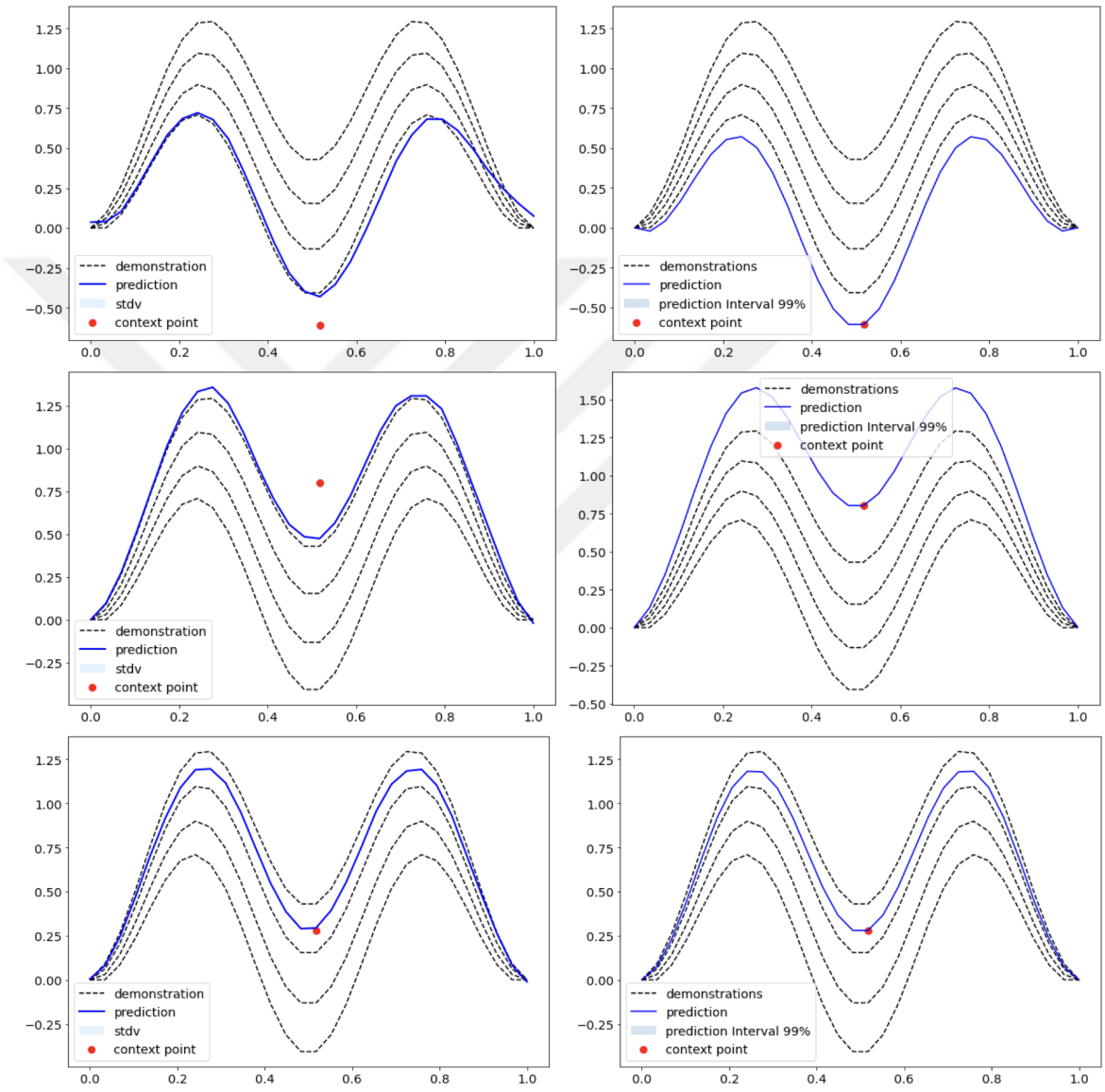


Figure 33: Comparison of estimated trajectories and confidence levels between CNMP (left) and CESN (right) models.

As illustrated in Figure 33, the CNMP model's estimated trajectory fails to adhere to the given context, and its confidence factor is unreliable. Despite the incorrect estimation, the CNMP model inaccurately conveys high confidence in its results. In

contrast, the CESN model not only adheres to the specified context but also accurately estimates the desired trajectory while maintaining a reliable confidence level in its predictions.

6.6.2 Multiple context points

The same process is repeated here, but this time using three context points instead of just one.

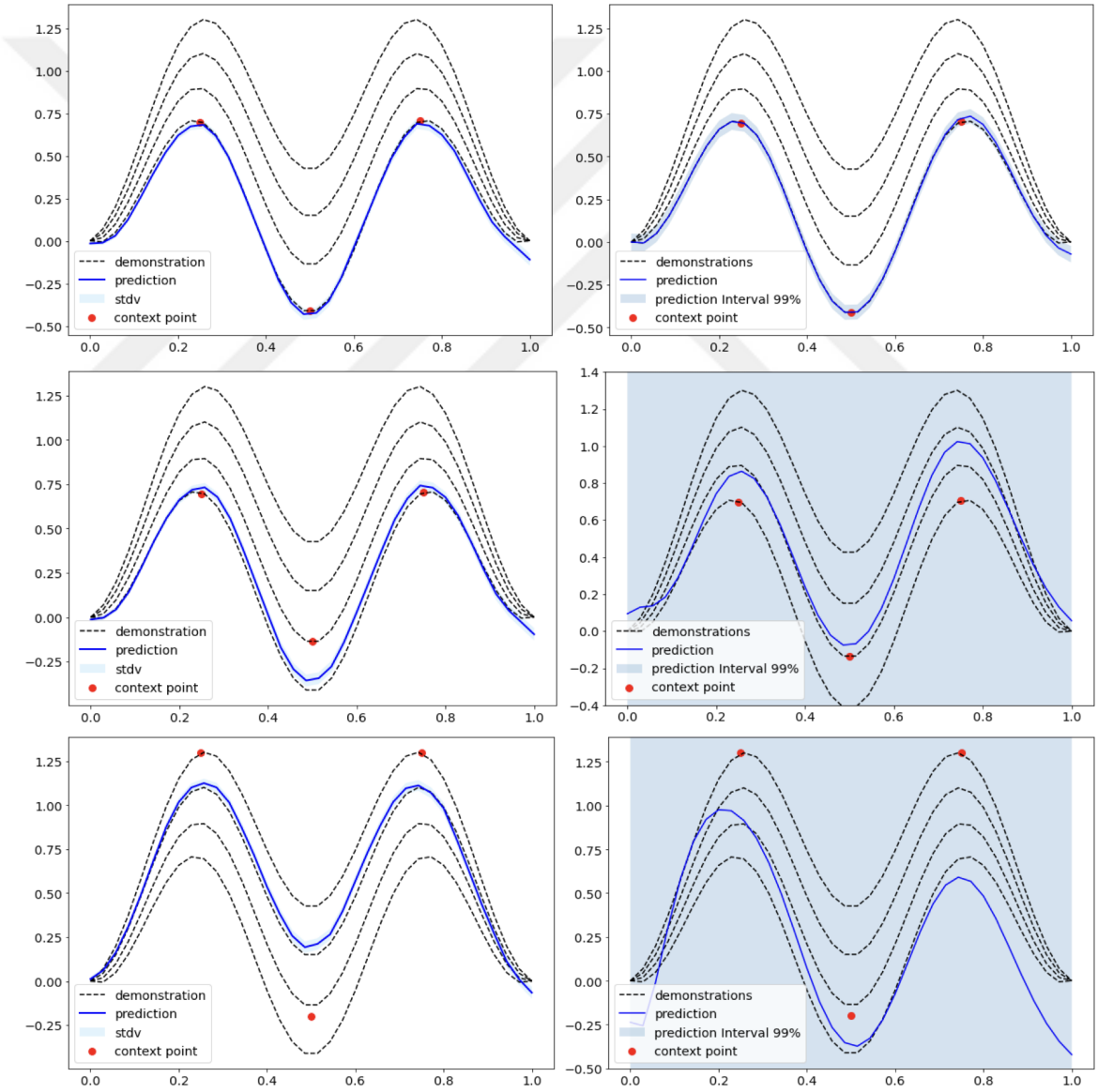


Figure 34: Comparison of estimated trajectories and confidence levels between CNMP (left) and CESN (right) models.

As illustrated in Figure 34, both models struggle to produce trajectories that align with the conditions specified by the context points. Despite this, the CNMP model exhibits a high level of confidence in its predictions, even though these predictions are inaccurate. In contrast, the CESN model demonstrates a very low level of confidence in its estimations, reflecting its recognition of the inaccuracies in its predictions.

This indicates that CESNs, with their ability to provide prediction intervals, exhibit promising performance in generating desired trajectories, even under extrapolation scenarios. Additionally, the confidence factors provided by CESNs are reliable, which makes CESNs a highly suitable and efficient choice for adaptive human-robot shared control systems. Their lightweight nature combined with robust performance in dynamic conditions affirms their potential for enhancing collaborative control tasks.

6.7 Conclusion

Context-based Echo State Networks (CESNs) represent a significant advancement offering a robust framework for generating movement primitives with high efficiency and adaptability. CESNs leverage context vectors to modulate reservoir dynamics, enabling the generation of varied movement patterns with a single linear readout weight vector. This design not only simplifies the training process but also enhances the real-time operational capability of robotic systems.

The effectiveness of CESNs is evident from their performance in various scenarios. Their computational efficiency is notable, as CESNs require minimal training time compared to other models, making them well-suited for real-time applications. Despite their simplicity, CESNs are capable of learning from a limited number of demonstrations and exhibit strong generalization capabilities for both interpolated and extrapolated contexts. Their robustness against control noise and delays further emphasizes their practical applicability.

In comparison to other methods such as CNMP, CESNs offer superior performance in generating desired trajectories, particularly in extrapolation cases. They provide reliable confidence intervals for predictions, reflecting the uncertainty in the estimations and making them a valuable tool for adaptive human-robot shared control systems. CESNs' ability to dynamically adjust their predictions based on context input and their efficient operation with minimal training highlight their potential for enhancing robotic applications.

Overall, CESNs present a promising solution for generating movement primitives and can significantly contribute to the development of adaptive and efficient robotic systems. Their lightweight nature, combined with their capacity to handle real-time dynamics and provide accurate predictions, positions them as an excellent choice for applications requiring both precision and adaptability.

CHAPTER VII

CONCLUSION AND FUTURE WORK

7.1 Summary of Findings

This thesis explores different aspects of human-robot learning and shared control, focusing on advancements that improve both human learning and system performance. The research highlights the potential of adaptive shared control frameworks to improve interactions between humans and robots, thereby boosting task efficiency and effectiveness.

Initially, the study demonstrated that adaptive shared control systems can significantly accelerate and enrich the learning process for human operators. By continuously adjusting to the inputs and intentions of the human, these systems foster a more intuitive and responsive interaction, which is crucial for complex tasks requiring a blend of human insight and robotic precision.

Expanding on this, the concept of integrating the 'robot's scope of knowledge' was explored. This approach proved valuable in enhancing task performance, especially in surgical applications. The system's ability to validate human intention estimation and adjust its role dynamically led to notable improvements in success rates, completion times, and obstacle avoidance. This integration highlights the effectiveness of combining robotic capabilities with human expertise for achieving optimal results.

The practical implementation of these adaptive shared control systems in medical suturing further demonstrated their benefits in critical environments. The system's real-time adjustment to human intentions played a key role in maintaining precision and reducing operator workload, showcasing the practical advantages of these frameworks in high-stakes tasks.

Additionally, the thesis addressed the limitations of Conditional Neural Processes (CNPs) in managing extrapolation issues by introducing reservoir computing. This alternative, with its lightweight and fast-training attributes, offers a promising solution for learning movement primitives, enhancing the adaptability and efficiency of shared control systems in dynamic and complex scenarios.

In summary, this research advances the field of human-robot interaction by illustrating how adaptive control systems, practical surgical applications, and innovative computational models work together to enhance human-robot collaborations. The findings underscore the transformative potential of these systems across various domains.

7.2 Future Work

Several directions for future research remain promising. Continued exploration and development in these areas could further advance the field and address existing challenges.

- One critical area for future work is the refinement and expansion of systems involving multiple robots working in conjunction with multiple human operators. Investigating how a group of humans and robots can be effectively coordinated in complex shared control scenarios presents both challenges and opportunities. Future research could focus on developing advanced algorithms and control strategies to enable seamless interaction among all parties, thereby optimizing their collaborative performance in intricate tasks and procedures.

- Another important direction is the advancement of online continual learning for robots. The ability of robots to adapt and improve their performance in real-time as they encounter new and evolving task requirements is crucial for their practical deployment. Future research could explore methods for integrating continual learning with existing shared control systems, allowing robots to

dynamically update their models and strategies based on ongoing interactions and feedback.

- Moreover, the continued exploration of LfD computational models to address challenges associated with adaptability in shared control systems, remains a vital area of research. Investigating how these models can be further optimized and integrated into adaptive control frameworks could lead to more robust and efficient systems.

CHAPTER VIII

MY EPILOGUE

The realm of human-robot interaction is set to transform in ways that may soon make our current achievements seem quaint. With the rise of adaptive and intelligent systems, the notion of robots merely assisting us will evolve into a partnership where they might anticipate our needs before we even articulate them. After all, who wouldn't want a robot that knows you need coffee before you realize you're about to pull an all-nighter? Or better yet, imagine a robot that switches your playlist from focus mode to a motivational anthem the second your productivity dips, no judgment, just seamless support.

As we move along this path, the line between human and robot may blur to the point where we forget who's leading and who's following. And maybe that's the point: a future where collaboration is so seamless it reshapes not just how we work, but how we think. It's not just about smarter tech; it's about solving problems with machines that are sharper, more perceptive, and maybe even as witty as we are.

Let's take it even further, even more thought provoking. Just a few years ago, there were hot discussions and concerns over machines "taking over human jobs," yet few would have anticipated that AI might soon be taking over the role of scientists themselves. And now, here we are, on the brink of a future where machines are not just automating lab work but contributing to scientific discovery. Critics argue that since AIs cannot think like humans, they will ultimately fall short. But our way of thinking is not the only path to knowledge. Perhaps machines, by thinking differently, will reveal insights and truth we might never have imagined.

REFERENCES

[1] M. Garnelo, D. Rosenbaum, C. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. Rezende, and S. A. Eslami, "Conditional neural processes," in *International Conference on Machine Learning*, pp. 1704–1713, PMLR, 2018.

[2] M. Y. Seker, M. Imre, J. H. Piater, and E. Ugur, "Conditional neural movement primitives.," in *Robotics: Science and Systems*, vol. 10, 2019.

[3] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Berlin: Springer, 2016.

[4] T. B. Sheridan, *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, 1992.

[5] D. A. Abbink, T. Carlson, M. Mulder, J. C. F. de Winter, F. Aminravan, T. L. Gibo, and E. R. Boer, "A topology of shared control: the framework for understanding the dynamic interaction between humans and automation," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 5, pp. 509–525, 2018.

[6] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: A survey," *Foundations and Trends in Human-Computer Interaction*, vol. 1, pp. 203–275, 2007.

[7] A. D. Dragan, K. C. T. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," in *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 301–308, 2013.

[8] T. Fong, C. Thorpe, and C. Baur, "Collaboration, dialogue, human-robot interaction," in *Proceedings of the 10th International Symposium on Robotics Research (ISRR)*, pp. 255–266, 2003.

[9] R. R. Murphy, "Human-robot interaction in rescue robotics," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 2, pp. 138–153, 2004.

[10] D. Feil-Seifer and M. J. Mataric, "Defining socially assistive robotics," in *Proceedings of the 9th International Conference on Rehabilitation Robotics*, pp. 465–468, 2005.

[11] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: A survey," *Foundations and Trends in Human-Computer Interaction*, vol. 13, no. 3, pp. 200–276, 2020.

[12] S. Natarajan and C. Breazeal, "Socially intelligent robots: Dimensions of human-robot interaction," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 159–182, 2022.

[13] X. Liu, W. Zhang, and L. Wang, "Enhancing human-robot interaction through machine learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 1, pp. 57–76, 2023.

[14] Z. Huang and Y. Hu, "Collaborative control in human-robot interaction: Principles and challenges," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 6, pp. 564–576, 2021.

[15] I. Mordatch and E. Todorov, "Reinforcement learning for robotic manipulation," *ICLR*, 2015.

[16] P. Kormushev, D. Kragic, and A. Billard, "Learning and generalization in robotics: A survey," *Autonomous Robots*, vol. 31, pp. 1–17, 2011.

[17] R. H. T. II, P. C. Lima, and J. H. Lee, "Medical robotics and computer-assisted surgery," *Annual Review of Biomedical Engineering*, vol. 15, pp. 137–165, 2013.

[18] L. Peternel, T. Petrič, E. Oztop, and J. Babič, "Teaching robots to cooperate with humans in dynamic manipulation tasks based on multi-modal human-in-the-loop approach," *Autonomous robots*, vol. 36, no. 1, pp. 123–136, 2014.

[19] R. Chipalkatty, G. Droge, and M. B. Egerstedt, "Less is more: Mixed-initiative model-predictive control with human inputs," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 695–703, 2013.

[20] D. Kushida, M. Nakamura, S. Goto, and N. Kyura, "Human direct teaching of industrial articulated robot arms based on force-free control," *Artificial Life and Robotics*, vol. 5, no. 1, pp. 26–32, 2001.

[21] R. Taylor, J. Smith, and E. Johnson, "Adaptive shared control in industrial robotics," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 2, pp. 567–580, 2022.

[22] M. Steele and R. B. Gillespie, "Shared control between human and machine: Using a haptic steering wheel to aid in land vehicle guidance," in *Proceedings of the human factors and ergonomics society annual meeting*, vol. 45, pp. 1671–1675, SAGE Publications Sage CA: Los Angeles, CA, 2001.

[23] A. Herweg, J. Gutzeit, S. Kleih, and A. Kübler, "Wheelchair control by elderly participants in a virtual environment with a brain-computer interface (bci) and tactile stimulation," *Biological psychology*, vol. 121, pp. 117–124, 2016.

[24] N. Hogan, "Impedance control: An approach to manipulation: Part i—theory," 1985.

[25] J. H. Palep, "Robotic assisted minimally invasive surgery," *Journal of minimal access surgery*, vol. 5, no. 1, p. 1, 2009.

[26] A. Bajcsy, D. P. Losey, M. K. O'Malley, and A. D. Dragan, "Learning robot objectives from physical human interaction," in *Conference on Robot Learning*, pp. 217–226, PMLR, 2017.

[27] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.

[28] N. Jarrassé, T. Charalambous, and E. Burdet, "A framework to describe, analyze and generate interactive motor behaviors," *PloS one*, vol. 7, no. 11, p. e49945, 2012.

[29] H. Wang, Q. Li, T. Zhang, and W. Chen, "Deep learning techniques for learning from demonstration in robotics," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 345–358, 2022.

[30] M. Ramadan and L. Zhao, "Shared control in teleoperation and assistive technologies: Recent advances and future directions," *IEEE Access*, vol. 10, pp. 22036–22049, 2022.

[31] P. Sharma and S. Lee, "Physiological signals for adaptive human-robot shared control," *IEEE Transactions on Biomedical Engineering*, vol. 70, no. 2, pp. 354–366, 2023.

[32] W. Jiang and H. Chen, "A bayesian framework for adaptive shared control in human-robot collaboration," *The International Journal of Robotics Research*, vol. 40, no. 8, pp. 1042–1055, 2021.

[33] M. Zhang and K. Li, "Augmented reality in adaptive shared control systems: Enhancing human-robot collaboration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 5, pp. 2102–2115, 2023.

[34] J. Lee and Y. Kim, "Improving shared control in human-robot interaction using augmented reality," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 4, pp. 284–297, 2021.

[35] M. Gonzalez, J. Hernandez, and L. Martinez, "Adaptive shared control for wheelchair navigation," *Journal of Rehabilitation Engineering*, vol. 34, no. 3, pp. 189–202, 2021.

[36] X. Wang, Y. Zhang, J. Li, and P. Liu, "Enhancing safety and comfort in autonomous vehicles through shared control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 1789–1803, 2022.

[37] G. Csibra, "Teleological and referential understanding of action in infancy," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 358, no. 1431, pp. 447–458, 2003.

[38] N. Amirshirzad, A. Kumru, and E. Oztop, "Human adaptation to human–robot shared control," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 2, pp. 126–136, 2019.

[39] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of intent inference for human-robot interaction," *Robotics and Autonomous Systems*, vol. 58, pp. 1151–1164, 2010.

[40] J. W. Crandall and M. A. Goodrich, "Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation," in *IEEE/RSJ international conference on intelligent robots and systems*, vol. 2, pp. 1290–1295, IEEE, 2002.

[41] W. Yu, R. Alqasemi, R. Dubey, and N. Pernalete, "Telemanipulation assistance based on motion intention recognition," in *Proceedings of the 2005 IEEE international conference on robotics and automation*, pp. 1121–1126, IEEE, 2005.

[42] D. Aarno, S. Ekvall, and D. Kragic, "Adaptive virtual fixtures for machine-assisted teleoperation tasks," in *Proceedings of the 2005 IEEE international conference on robotics and automation*, pp. 1139–1144, IEEE, 2005.

[43] D. Nicolis, A. M. Zanchettin, and P. Rocco, "Human intention estimation based on neural networks for enhanced collaboration with robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1326–1333, IEEE, 2018.

[44] Y. Sun, W. Wang, Y. Chen, and Y. Jia, "Learn how to assist humans through human teaching and robot learning in human 2013;robot collaborative assembly," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 2, pp. 728–738, 2022.

[45] M.-L. Lee, W. Liu, S. Behdad, X. Liang, and M. Zheng, "Robot-assisted disassembly sequence planning with real-time human motion prediction," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 1, pp. 438–450, 2023.

[46] C. Huang, T. Geng, H. Zhang, L. Yang, B. Gao, G. Wang, and Z. Jin, "Human intention estimation in human-robot collaboration: a machine learning approach," *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102125, 2022.

[47] Z. Yang and D. Wu, "Human intention estimation in human-robot interaction using deep learning and gaussian processes," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 3, pp. 1886–1896, 2023.

[48] J. Butepage, M. J. Black, D. Kragic, and H. Kjellstrom, "Deep representation learning for human motion prediction and classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6158–6166, 2017.

[49] A. C. Tsitos, M. Dagioglou, and T. Giannakopoulos, "Real-time feasibility of a human intention method evaluated through a competitive human-robot reaching game," in *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '22, p. 1080–1084, IEEE Press, 2022.

[50] W. Wang, R. Li, Y. Chen, and Y. Jia, "Human intention prediction in human-robot collaborative tasks," in *Companion of the 2018 ACM/IEEE international conference on human-robot interaction*, pp. 279–280, 2018.

[51] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malley, "A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction," *Applied Mechanics Reviews*, vol. 70, no. 1, p. 010804, 2018.

[52] Z. Huang, Y.-J. Mun, X. Li, Y. Xie, N. Zhong, W. Liang, J. Geng, T. Chen, and K. Driggs-Campbell, "Hierarchical intention tracking for robust human-robot collaboration in industrial assembly tasks," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9821–9828, 2023.

[53] K. Muelling, H. Kim, M. Kuderer, and W. Burgard, "Learning dynamic context for intention prediction in human-robot collaboration," *The International Journal of Robotics Research*, vol. 40, no. 1, pp. 63–81, 2021.

[54] J. Li, X. Zhang, Q. Zhang, and H. Liu, "Multimodal data fusion for human intention estimation in human-robot interaction," *Robotics and Autonomous Systems*, vol. 151, p. 103916, 2023.

[55] N. Amirshirzad, O. Kaya, and E. Oztop, "Synergistic human-robot shared control via human goal estimation," in *2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 691–695, IEEE, 2016.

[56] N. Amirshirzad, E. Ugur, O. Bebek, and E. Oztop, "Adaptive shared control with human intention estimation for human agent collaboration," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pp. 653–658, IEEE, 2022.

[57] J. Su, "Representation and inference of user intention for internet robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 8, pp. 995–1002, 2014.

[58] A. Franchi, C. Secchi, M. Ryll, H. H. Bulthoff, and P. R. Giordano, "Shared control: Balancing autonomy and human assistance with a group of quadrotor uavs," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 57–68, 2012.

[59] H. Boessenkool, D. A. Abbink, C. J. Heemskerk, F. C. van der Helm, and J. G. Wildenbeest, "A task-specific analysis of the benefit of haptic shared control during telemanipulation," *IEEE Transactions on Haptics*, vol. 6, no. 1, pp. 2–12, 2012.

[60] Y. Duan and M. Zhao, "Learning from demonstration: Recent advances and challenges," *Robotics and Autonomous Systems*, vol. 140, p. 103855, 2022.

[61] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 286–298, 2007.

[62] M. Hersch, F. Guenter, S. Calinon, and A. Billard, "Dynamical system modulation for robot learning via kinesthetic demonstrations," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1463–1467, 2008.

[63] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 297–330, 2020.

[64] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4693–4700, IEEE, 2018.

[65] J. Schulman, J. Ho, C. Lee, and P. Abbeel, "Learning from demonstrations through the use of non-rigid registration," in *Robotics Research*, pp. 339–354, Springer, 2016.

[66] X. Liu, W. Zhang, and L. Wang, "Learning from demonstration with reinforcement learning for robotic manipulation tasks," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 112–125, 2023.

[67] J. Park, J. Lee, and S.-H. Kim, "Design principles for interface design in human-robot interaction," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 5, pp. 401–414, 2021.

[68] Y. Xu, Y. Sun, and D. Wang, "Augmented reality interfaces for human-robot collaboration: A review," *International Journal of Human-Computer Interaction*, vol. 39, no. 4, pp. 293–307, 2023.

[69] M. Kim, J. Park, and J. Lee, "User-centered design approaches for human-robot interaction," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 2, pp. 137–150, 2022.

[70] H. Choi, E. Lee, and S.-H. Kim, "Iterative design and testing of interfaces in human-robot interaction," *Journal of Mechanical Science and Technology*, vol. 37, no. 1, pp. 63–76, 2023.

[71] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[72] S. Schaal, "Dynamic movement primitives-a framework for motor control in humans and humanoid robotics," *Adaptive motion of animals and machines*, pp. 261–280, 2006.

[73] R. Huang, H. Cheng, H. Guo, Q. Chen, and X. Lin, "Hierarchical interactive learning for a human-powered augmentation lower exoskeleton," in *2016 IEEE international conference on robotics and automation (ICRA)*, pp. 257–263, IEEE, 2016.

[74] A. Gams, B. Nemec, A. J. Ijspeert, and A. Ude, "Coupling movement primitives: Interaction with the environment and bimanual tasks," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 816–830, 2014.

[75] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," *Autonomous Robots*, vol. 42, no. 3, pp. 529–551, 2018.

[76] H. G. Sung, *Gaussian mixture regression and classification*. Rice University, 2004.

[77] D. A. Reynolds, "Gaussian mixture models.," *Encyclopedia of biometrics*, vol. 741, pp. 659–663, 2009.

[78] G. P. Mylonas, P. Giataganas, M. Chaudery, V. Vitiello, A. Darzi, and G.-Z. Yang, "Autonomous efast ultrasound scanning by a robotic manipulator using learning from demonstrations," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3251–3256, IEEE, 2013.

[79] P. Giataganas, V. Vitiello, V. Simaiaki, E. Lopez, and G.-Z. Yang, "Cooperative in situ microscopic scanning and simultaneous tissue surface reconstruction using a compliant robotic manipulator," in *2013 IEEE International Conference on Robotics and Automation*, pp. 5378–5383, IEEE, 2013.

[80] J. Bernardo, J. Berger, A. Dawid, A. Smith, *et al.*, "Regression and classification using gaussian process priors," *Bayesian statistics*, vol. 6, p. 475, 1998.

[81] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," *Advances in neural information processing systems*, vol. 18, p. 1257, 2006.

[82] M. Garnelo, J. Schwarz, D. Rosenbaum, F. Viola, D. J. Rezende, S. Eslami, and Y. W. Teh, "Neural processes," *arXiv preprint arXiv:1807.01622*, 2018.

[83] H. Kim, A. Mnih, J. Schwarz, M. Garnelo, A. Eslami, D. Rosenbaum, O. Vinyals, and Y. W. Teh, "Attentive neural processes," *arXiv preprint arXiv:1901.05761*, 2019.

[84] M. T. Akbulut, E. Oztop, M. Y. Seker, H. Xue, A. E. Tekden, and E. Ugur, "Acnmp: Skill transfer and task extrapolation through learning from demonstration and reinforcement learning via representation sharing," *arXiv preprint arXiv:2003.11334*, 2020.

[85] J. Murphy, D. Smith, and S. Johnson, "Testing and validation of adaptive shared control systems," *Journal of Robotics and Automation*, vol. 45, no. 2, pp. 123–135, 2023.

[86] N. Amirshirzad, "Human-robot collaboration for synergistic task execution," Master's thesis, 2017.

[87] S. Kumar, C. Savur, and F. Sahin, "Survey of human–robot collaboration in industrial settings: Awareness, intelligence, and compliance," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 280–297, 2021.

[88] Y. Torres, S. Nadeau, and K. Landau, "Classification and quantification of human error in manufacturing: a case study in complex manual assembly," *Applied Sciences*, vol. 11, no. 2, p. 749, 2021.

[89] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.

[90] A. D. Dragan, "Robot planning with mathematical models of human state and action," *arXiv preprint arXiv:1705.04226*, 2017.

[91] M. A. Zamani and E. Oztop, "Simultaneous human-robot adaptation for effective skill transfer," in *2015 International Conference on Advanced Robotics (ICAR)*, pp. 78–84, 2015.

[92] H. Maske, E. Kieson, G. Chowdhary, and C. Abramson, "Can co-robots learn to teach?," *arXiv preprint arXiv:1611.07490*, 2016.

[93] M. Allain, S. Konduri, H. Maske, P. R. Pagilla, and G. Chowdhary, "Blended shared control of a hydraulic excavator," *IFAC-papersonline*, vol. 50, no. 1, pp. 14928–14933, 2017.

[94] S. C. Jugade, A. C. Victorino, and V. B. Cherfaoui, "Human-intelligent system shared control strategy with conflict resolution," in *2018 IEEE 14th International Conference on Control and Automation (ICCA)*, pp. 686–691, IEEE, 2018.

[95] F. Vanderhaegen, "Heuristic-based method for conflict discovery of shared control between humans and autonomous systems-a driving automation case study," *Robotics and Autonomous Systems*, vol. 146, p. 103867, 2021.

[96] W. Li, Q. Li, S. E. Li, R. Li, Y. Ren, and W. Wang, "Indirect shared control through non-zero sum differential game for cooperative automated driving," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[97] M. Imre, E. Oztop, Y. Nagai, and E. Ugur, "Affordance-based altruistic robotic architecture for human–robot collaboration," *Adaptive Behavior*, vol. 27, no. 4, pp. 223–241, 2019.

[98] S. Schaal, "Learning from demonstration," *Advances in neural information processing systems*, vol. 9, 1996.

[99] E. Rohmer, S. P. N. Singh, and M. Freese, "Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework," 2013. www.coppeliarobotics.com.

[100] E. Rohmer, S. P. N. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1321–1326, 2013.

[101] R. G. O'Brien and M. K. Kaiser, "Manova method for analyzing repeated measures designs: an extensive primer.," *Psychological bulletin*, vol. 97, no. 2, p. 316, 1985.

[102] K. P. Weinfurt, *Repeated measures analysis: ANOVA, MANOVA, and HLM.*, pp. 317–361. Reading and understanding MORE multivariate statistics., Washington, DC, US: American Psychological Association, 2000.

[103] A. M. Okamura, C. Simone, and M. K. O'Malley, "Shared control in haptic systems for performance enhancement and training," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, no. 1, pp. 75–85, 2010.

[104] J. J. Abbott and A. M. Okamura, "Surgical robotics: Enabling technology for minimally invasive surgery," *IEEE Robotics & Automation Magazine*, vol. 14, no. 3, pp. 21–30, 2007.

[105] T. Lane, "A short history of robotic surgery," *The Annals of The Royal College of Surgeons of England*, vol. 100, no. 6_sup, pp. 5–7, 2018.

[106] S. DiMaio, M. Hanuschik, and U. Kreaden, "The da vinci surgical system," in *Surgical Robotics*, pp. 199–217, Springer, 2011.

[107] A. M. Okamura, M. J. Matarić, and H. I. Christensen, "Medical and health-care robotics," *IEEE Robotics Automation Magazine*, vol. 17, no. 3, pp. 26–37, 2010.

[108] R. H. Taylor and D. Stoianovici, "Medical robotics and computer-integrated surgery," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, pp. 765–781, 2003.

[109] N. Simaan, D. Yuh, C. E. Reiley, and R. H. Taylor, "Design and integration of a telerobotic system for minimally invasive surgery of the throat," *International Journal of Robotics Research*, vol. 28, no. 9, pp. 1134–1153, 2009.

[110] O. Bohigas, M. Manubens, and L. Ros, "M2r2: A multi-arm and multi-camera robotic assistant for surgery," *IEEE Robotics & Automation Magazine*, vol. 20, no. 4, pp. 96–106, 2013.

[111] A. Kapoor, M. Li, and R. H. Taylor, "Spatial motion constraints for robot assisted suturing using virtual fixtures," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005* (J. S. Duncan and G. Gerig, eds.), (Berlin, Heidelberg), pp. 89–96, Springer Berlin Heidelberg, 2005.

[112] A. Abiri, S. J. Askari, A. Tao, Y. Y. Juo, Y. Dai, J. Pensa, R. Candler, E. P. Dutson, and W. S. Grundfest, "Suture breakage warning system for robotic surgery," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 4, pp. 1165–1171, 2019.

[113] T. Liu and M. C. Cavusoglu, "Needle grasp and entry port selection for automatic execution of suturing tasks in robotic minimally invasive surgery," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 552–563, 2016.

[114] A. Shademan, R. S. Decker, J. D. Opfermann, S. Leonard, A. Krieger, and P. C. W. Kim, "Supervised autonomous robotic soft tissue surgery," *Science Translational Medicine*, vol. 8, no. 337, pp. 337ra64–337ra64, 2016.

[115] S. Sen, A. Garg, D. V. Gealy, S. McKinley, Y. Jen, and K. Goldberg, "Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4178–4185, 2016.

[116] S. Iyer, T. Looi, and J. Drake, "A single arm, single camera system for automated suturing," in *2013 IEEE International Conference on Robotics and Automation*, pp. 239–244, 2013.

[117] S. Leonard, A. Shademan, Y. Kim, A. Krieger, and P. C. W. Kim, "Smart tissue anastomosis robot (star): Accuracy evaluation for supervisory suturing using near-infrared fluorescent markers," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1889–1894, 2014.

[118] C. D'Ettorre, G. Dwyer, X. Du, F. Chadebecq, F. Vasconcelos, E. De Momi, and D. Stoyanov, "Automated pick-up of suturing needles for robotic surgical assistance," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1370–1377, 2018.

[119] N. Hogan, H. I. Krebs, B. Rohrer, J. Palazzolo, L. Dipietro, S. E. Fasoli, J. Stein, R. E. Hughes, and W. R. Frontera, "Control of robotic rehabilitation devices: a review," *IEEE Transactions on Robotics*, vol. 10, no. 3, pp. 248–257, 2015.

[120] S. DiMaio, M. Hanuschik, and U. Kreaden, "The da vinci surgical system," in *Surgical robotics*, pp. 199–217, Springer, 2011. Accessed: Feb. 1, 2021.

[121] B. Sunal, "Tele-operation and control of robotic surgical assistants for suturing task," Master's thesis, 2022.

[122] "Franka ROS repository." Accessed: Feb. 1, 2021.

[123] ROBOTIS, "e-manual xl320." Accessed: Feb. 1, 2021.

[124] B. Sunal, E. Oztop, and O. Bebek, "Adaptive inverse kinematics of a 9-dof surgical robot for effective manipulation," in *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*, pp. 678–683, 2019.

[125] N. Amirshirzad, B. Sunal, O. Bebek, and E. Oztop, "Learning medical suturing primitives for autonomous suturing," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pp. 256–261, IEEE, 2021.

[126] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning movement primitives," in *Robotics Research. The Eleventh International Symposium: With 303 Figures*, pp. 561–572, Springer, 2005.

[127] H. Jaeger, "Echo state network," *scholarpedia*, vol. 2, no. 9, p. 2330, 2007.

[128] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, pp. 64–67, 2001.

[129] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer science review*, vol. 3, no. 3, pp. 127–149, 2009.

[130] J. Tani, M. Ito, and Y. Sugita, "Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using rnnpb," *Neural Networks*, vol. 17, no. 8-9, pp. 1273–1289, 2004.

[131] L. Annabi, A. Pitti, and M. Quoy, "Autonomous learning and chaining of motor primitives using the free energy principle," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2020.

[132] A. F. Krause, B. Bläsing, V. Dürr, and T. Schack, "Direct control of an active tactile sensor using echo state networks," *Human Centered Robot Systems: Cognition, Interaction, Technology*, pp. 11–21, 2009.

[133] M. Lukoševičius, "A practical guide to applying echo state networks," *Neural Networks: Tricks of the Trade: Second Edition*, pp. 659–686, 2012.

[134] N. Amirshirzad, M. Asada, and E. Oztop, "Context based echo state networks for robot movement primitives," in *2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1077–1082, IEEE, 2023.

[135] F. Triefenbach, A. Jalalvand, B. Schrauwen, and J.-P. Martens, "Phoneme recognition with large hierarchical reservoirs," *Advances in neural information processing systems*, vol. 23, 2010.

[136] Franka Emika, "Panda solution." Accessed: Feb. 1, 2021.

[137] "Gazebo sim." Accessed: Feb. 1, 2021.