LEARNING MULTI-STEP DUAL-ARM TASKS FROM DEMONSTRATIONS

by

Natalia Sanchez-Tamayo

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science in Industrial Engineering

School of Industrial Engineering

West Lafayette, Indiana

August 2020

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Juan Wachs, Chair

School of Industrial Engineering

Dr. Ramses Martinez

School of Industrial Engineering

Dr. Yexiang Xue

Department of Computer Science

Approved by:

Dr. Abhijit Deshmukh

To my parents, my sister, family, and friends

ACKNOWLEDGEMENTS

I would like to acknowledge the invaluable support and assistance from all the people who have helped shape my professional and personal life during my graduate studies at Purdue University. Foremost, I would like to thank my parents and my sister. Their endless encouragement and support throughout my life has motivated me to pursue my interests and ambitions. I will always appreciate their thoughtful advice and joyful company.

I would like to express my sincere gratitude to my advisor, Prof. Juan P. Wachs, for his continuous support during my graduate studies. His patience and motivation encouraged me to continuously push forward and aim for the best. His constant and thorough feedback in all aspects of research helped me become a better researcher. I would also like to thank the rest of my committee, Yexiang Xue and Ramses Martinez, for their insightful comments and suggestions on this thesis and their support throughout my time at Purdue.

To the Arlinghaus family for acting as my family in the United States. Their support during the last months of my graduate program brought happiness to my daily life and eased the difficulties produced by the Covid-19 pandemic. A special acknowledgement to Ryan Arlinghaus, for his outstanding support during my studies, for promoting stimulating discussions and motivating spontaneous laughs.

A special thanks to my lab mates at the Intelligent Systems and Assistive Technologies laboratory. To Maru Cabrera, Glebys Gonzalez and Naveen Madapana, I thank you all for being great mentors and awesome friends, for the hilarious conversations, and for insightful discussions and advice. To Juan Antonio, Ting, Oscar, Marina, Tian, Anirudh, Daniela, Edgar, Xinguang and Chenxi, I thank you for making the laboratory and Purdue an awesome place to work.

I gratefully acknowledge the founding sources that supported me throughout my studies at Purdue. During my graduate studies, I have been supported by the Office of the Assistant Secretary of Defense for Health Affairs under Awards No. W81XWH-14-1-0042 and No. W81XWH-18-1-0769. Some parts of the work presented in this thesis will be submitted to journal publications for publishing.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	Application programming interface
BC	Behavioral Cloning
CNN	Convolutional Neural Network
CUSUM	Cumulative Sum
DIL	Deep Imitation Learning
DOF	Degrees Of Freedom
DRL	Deep Reinforcement Learning
DTW	Dynamic Time Warping
dVRK	da Vinci Research kit
FLS	Fundamentals of Laparoscopic Surgery
FSM	Finite State Machine
HMD	Head Mounted Display
HMM	Hidden Markov Models
IL	Imitation learning
IRL	Inverse Reinforcement Learning
LSTM	Long-Short Term Memory
MIS	Minimally Invasive Surgery
MRRP	Microsurgical robot research platform
OR	Operating Room
RAS	Robotic-Assisted Surgery
RL	Reinforcement Learning
RNN	Recurrent Neural Networks
ROS	Robot Operating System
VR	Virtual Reality

ABSTRACT

Surgeon expertise can be difficult to capture through direct robot programming. Deep imitation learning (DIL) is a popular method for teaching robots to autonomously execute tasks through learning from demonstrations. DIL approaches have been previously applied to surgical automation. However, previous approaches do not consider the full range of robot dexterous motion required in general surgical task, by leaving out tooltip rotation changes or modeling one robotic arm only. Hence, they are not directly applicable for tasks that require rotation and dual-arm collaboration such as debridement. We propose to address this limitation by formulating a DIL approach for the execution of dual-arm surgical tasks including changes in tooltip orientation, position and gripper actions.

In this thesis, a framework for multi-step surgical task automation is designed and implemented by leveraging deep imitation learning. The framework optimizes Recurrent Neural Networks (RNNs) for the execution of the whole surgical tasks while considering tooltip translations, rotations as well as gripper actions. The network architecture proposed implicitly optimizes for the interaction between two robotic arms as opposed to modeling each arm independently. The networks were trained directly from the human demonstrations and do not require to create task specific hand-crafted models or to manually segment the demonstrations.

The proposed framework was implemented and evaluated in simulation for two relevant surgical tasks, the peg transfer task and the surgical debridement. The tasks were tested under random initial conditions to challenge the robustness of the networks to generalize to variable settings. The performance of the framework was assessed using task and subtask success as well as a set of quantitative metrics. Experimental evaluation showed favorable results for automating surgical tasks under variable conditions for the surgical debridement, which obtained a task success rate comparable to the human task success. For the peg transfer task, the framework displayed moderate overall task success. Quantitative metrics indicate that the robot generated trajectories possess similar or better motion economy that the human demonstrations.

1. INTRODUCTION

Robotic Assisted Surgery (RAS) has impacted the surgical field drastically. As of 2019, four million procedures have been completed with the da Vinci Surgical Systems utilizing more than 4,000 surgical systems installed around the world (Miller & Curet, 2019). RAS has advanced minimally invasive surgery by enhancing the view of the surgical field, providing additional dexterity and precision as well as clinical benefits such as decreased blood loss, reduced procedure time and hospital stay (Mukherjee & Sinha, 2020).

In MIS, the automation of surgical procedures can provide benefits such as reducing surgeon fatigue over long operations (Hwang et al., 2020; Watanabe et al., 2016) as well as assisting the surgeon on repetitive tasks such as suturing (Dolph et al., 2019). Surgical automation can also be beneficial for long-distance robot teleoperated surgery that can be vulnerable to cyberattacks (Bonaci et al., 2015) and often exhibit a limited bandwidth, latency, and even loss of signal (Marescaux et al., 2002). In these cases, automation could potentially be used to take over the surgical task until reliable communication is restored.

Surgical tasks are often multi-step containing several steps or primitive actions that can be repeated within the task. For example, suturing involves several primitive actions such as inserting the needle in the skin, running the suture through the tissue, and pulling the thread tight, among others. Previous advances in task automation have primarily focused on creating models for specific surgical tasks; such as tumor palpation (Garg et al., 2016), debridement (B. Kehoe et al., 2014; Richter et al., 2019; Seita et al., 2018), suturing (Sen et al., 2016) bowel anastomosis (Shademan et al., 2016) and blunt dissection (Nagy & Haidegger, 2019). These works generally rely on human expert knowledge to model suitable robot actions or predefined movements, for example generating trajectories parametrized according to the specific task or its constituent primitive actions. These models do not generalize to new tasks without requiring new expert knowledge. Our work aims at learning full task automation directly from human demonstrations without the need of creating individual models for each task or its primitive actions. The expert knowledge is inherently learned from the demonstrations allowing to model distinct tasks and bypassing the need of hand-crafted models for task execution.

With the advent of deep learning, intelligent systems can potentially be trained to learn implicit models for surgical task execution directly from human demonstrations. Deep learning approaches have been employed to learn full tasks from demonstrations in tissue manipulation (Shin et al., 2019) and autonomous tool navigation inside the eye (J. W. Kim et al., 2019). These works model tooltip position throughout the surgical task but do not consider changes in tooltip orientation. In order to replicate human skills in surgical tasks such as knot-tying, modeling tooltip orientations as well as its position, is required. Therefore, we leverage on deep learning approaches to model the complexity of surgical tasks from human demonstrations considering dexterous tooltip motion as well as dual-arm interaction.

We leverage deep learning to model task execution directly from expert demonstrations. This is known as Deep Imitation Learning (DIL). DIL employs human demonstrations to approximate robot control policies that predict the robot actions (e.g. joint commands) based on a set of observations (e.g. current joint states, state of the environment). Learning such policies from human expert demonstrations can be beneficial to learn models for task execution that implicitly learn user preferences or task constraints that cannot be easily programmed.

This thesis presents a framework that leverages DIL and human demonstrations to train robot control policies for the execution of distinct multi-step surgical tasks. The framework optimizes Recurrent Neural Networks (RNNs) to model robot tooltip position, orientation, and gripper actions while implicitly optimizing for the interaction between two robotic arms in multilateral manipulation. Such an approach offers a holistic framework to learn dual-arm robot control policies for surgical tasks from human demonstrations. The RNNs can be trained over the data of the full task, allowing the network to implicitly learn full task execution. Thus, eliminating the need to model individual primitive actions for a specific task. The proposed framework can also be applied to different surgical tasks by training the robot control policy over the dataset of the new task.

First, a dataset for learning two tasks pertinent to surgical robotics is built. This is done by creating an intuitive system for data collection from human demonstrations. A simulation environment and virtual reality technologies are used for this purpose. The proposed method for data collection aims to capture significant variability in the setting by embedding randomization in the environment's

initial state. For instance, by modifying the starting position and orientation of the pegboard in a peg transfer task. Such variation increases robustness in the imitation learning algorithm and improves the applicability of the framework to test scenarios with highly variable initial conditions.

Our framework for task learning leverages commonalities between task demonstrations. This allows us to map different variations from an original setting to a coordinate space representation where control policies are learned more efficiently. The proposed framework uses spatial knowledge of task variation, such as environment rotations, scale, and translations to represent user demonstrations in a unified coordinate space where the demonstrated trajectories share common traits, such as the start and the goal positions of the target object. This representation is used to train RNNs with fewer examples, as one example in the unified space can represent different spatial variations in the original robot setting.

Our approach is validated through experiments comparing robot to human performance during task execution in two surgical tasks, surgical debridement and peg transfer task. The performance of the networks is determined by the full task and subtask success as well as quantitative metrics over the generated trajectories related to task completion time and economy of motion.

1.1 Problem statement.

Design and implement a framework that leverages deep learning and human demonstrations to learn dual-arm tasks in the context of surgical robotics. Such framework includes modeling different surgical tasks with multiple steps in a generalizable manner.

1.1.1 Research questions

RQ1: How can deep imitation learning be used to learn multi-step tasks in the context of surgical robotics?

This question relates to the development and implementation of a learning framework for the execution of a dexterous surgical task.

RQ2: How to integrate dual-arm task learning for deep imitation learning?

This question pertains to the formulation of an architecture to train robot control policies for dual-arm task execution.

RQ3: How to evaluate the success of the proposed framework?

This question refers to the validation, metrics, and assessment of the proposed framework by evaluating the performance of the learned policies in a simulated environment for two distinct surgical tasks.

1.2 Thesis structure

The remainder of this thesis is structured as follows. Chapter 2 presents a literature review that summarizes previous related work. Chapter 3 describes the methodology applied to the problem of autonomous execution of surgical tasks. An overview of the learning framework is described with a description of the recurrent neural network architecture, data processing, and training methodology. The system for data collection and testing of task execution are described as well. In Chapter 4, the experiments and results are described and discussed. Finally, Chapter 5 discusses the conclusions and future work.

1.3 Summary

This section introduced the motivation for researching deep imitation learning approaches for surgical task execution. Then, an introduction of our proposed framework is presented. At last, three research questions are described: the first explores imitation learning frameworks for the execution of surgical tasks. The second question explores the integration of dual-arm task learning in deep imitation learning. The third poses the metrics and experiments to be conducted to evaluate the framework.

2. LITERATURE REVIEW

This chapter is a review of the previous and ongoing research on the main areas of learning and autonomy in surgical robotics. First, a background on major areas of learning through examples in robotics is presented, including imitation learning and reinforcement learning. Once the teaching techniques are covered, the main domain is discussed, which is surgical robotics. Research platforms and industrial settings related to surgical robotics are presented, and the extension of 3D printing technologies to enhance existing robots of this kind. Then, autonomous and semi-autonomous surgical task execution is discussed. At last, deep learning approaches for imitation learning in robotic manipulation are presented.

2.1 Online and offline learning

Offline learning refers to the family of machine learning methods where a learning algorithm is trained with a collection of data that is given in advance (Aroussi & Mellouk, 2014). Once the model is trained over the entire dataset, it can be used to make predictions over new data. An example of offline learning is a neural network for image recognition that is trained on the previously available dataset ImageNet (Deng et al., 2009). The trained network can then be used to predict image labels in entirely new settings.

In contrast, online learning refers to the methods that train a model from a sequence of data instances one by one at each time (Hoi et al., 2018). Thus, online learning focuses on the incremental acquisition of knowledge. For example, a commonly studied task in online learning is spam email filtering (W. Liu & Wang, 2012), where the model is continuously being updated as new information becomes available.

In this thesis, a learning framework for robot task execution from human demonstrations is presented. Our approach employs human demonstrations collected in advance. Therefore, it falls within the category of offline machine learning approaches.

2.2 Imitation learning in robotics

Imitation learning (IL) is referred to as the research area that focuses on learning techniques from demonstration, usually in the form of examples, provided by a human teacher (Osa, Pajarinen, et al., 2018). Imitation learning is also referred to as learning from demonstration (Atkeson & Schaal, 1997). The basic idea of IL involves developing algorithms that can learn a behavior through demonstrations without explicitly programming such behavior. IL is a compelling approach for applications where the ideal behavior cannot be easily defined as an optimization problem but can be demonstrated (Ravichandar et al., 2020). Because the policies are learned directly from human demonstrations, IL has been found to implicitly learn user preference as well as task constraints that can enable adaptive behavior for unstructured environments (Kroemer et al., 2019; Ravichandar et al., 2020).

The objective of IL consists of learning an underlying teacher policy that is unknown, except for (usually sparse and noisy) demonstrations that are presented as training examples (Chernova & Thomaz, 2014). Such demonstrations can be encoded as a set of state-action pairs, where the state can be defined, for example, as the current position of the robot, the state and velocity of the robot joints and the position of objects of interest (Hussein et al., 2017). Similarly, the corresponding action describes the control output, that can be represented as motor actions such as joint states, or higher-level instructions such as “pick up an object”. Imitation learning research interest has been steadily increasing over the past decade, being used for numerous applications in robotics (Ravichandar et al., 2020). IL has been used in autonomous vehicles, (Abbeel et al., 2010), specifically in helicopter acrobatic maneuvers by learning a library of short trajectories and using a high-level planner to concatenate such trajectories together into a sequence. For autonomous driving and aerial vehicles, IL has been used to create end-to-end controllers (Abdou et al., 2019; Bojarski et al., 2016; Loquercio et al., 2018). In surgical robotics, IL has been adopted for learning surgical tasks from teleoperated demonstrations, such as knot-tying (Berg et al., 2010; Mayer et al., 2006), pattern cutting, and debridement (Murali et al., 2015).

Behavioral Cloning (BC), is a category of IL approaches that focusses on approximating a function to directly map input states to output actions (Ravichandar et al., 2020). In imitation learning, this function is called a policy, which can be learned, for example, using supervised learning approaches trained on a dataset of state-action pairs.

Inverse Reinforcement Learning (IRL) refers to a different category of IL methods that try to infer an underlying reward function from the demonstrations instead of learning a direct policy (Kroemer et al., 2019). Once learned, the reward function can be employed to compute optimal actions in classical reinforcement learning approaches.

Deep Imitation Learning (DIL) refers to methods that train neural networks from demonstration to learn expert control policies (Ericson, 2018). In the literature, deep imitation learning generally refers to networks trained using supervised learning methods (Yu et al., 2020). For example, DIL can use behavioral cloning to train a neural network with a fixed dataset of state-action pairs collected from demonstrations (Kuefler et al., 2017). DIL has been used for several applications in robotics to train neural network control policies from human demonstrations (Aşık et al., 2018; J. W. Kim et al., 2019; Yuejiang Liu et al., 2018; Saleh et al., 2018).

In DIL, the number of collected demonstrations can affect the success of the trained policies. (Nagy & Haidegger, 2019; T. Zhang et al., 2018) found that increasing the human demonstrations shows an improvement in the task execution success. When data is scarce, the observations can be generated artificially. For example, automatically generated trajectories have been proposed to bypass the problem of data acquisition in DIL (J. W. Kim et al., 2019). They have artificially generated over 2000 examples for the navigation of a surgical tool in the eye. (James et al., 2017b) have generated 4000 examples to solve a pick and place task. In addition, data augmentation techniques have been proposed to facilitate learning of manipulation tasks based on original demonstrations collected by a human user. For example, when learning from trajectories, expanding the initialization position of trajectories (J. W. Kim et al., 2019), and adding artificial shifts and rotations to the demonstrations (Bojarski et al., 2016; Rahmatizadeh et al., 2016) have been applied for data augmentation.

However, deciding on what type of transformations should be applied to augment the data in a meaningful manner, is challenging. Particularly, when the demonstrations represent both robot position and orientation, data augmentation should consider transformations that convey meaningful shifts, rotations, and scaling. Moreover, the choice of specific gripper orientation representation can affect the data efficiency of the model used to predict end-effector pose. Previous work by (Y. Zhou et al., 2019) shows that representing orientations (e.g. Euler,

quaternion) in an equivalent space (e.g. such as 6D rotation representations) facilitates training of neural networks. These findings suggest that alternative representations of pose trajectory demonstrations could potentially facilitate learning in neural networks for multiple applications. In our work, we leverage transformations commonly used for artificial data augmentation (e.g shifts, rotations, translations) to represent the human demonstrations in an equivalent space that facilitates learning of neural network control policies.

2.3 Reinforcement learning

Reinforcement learning (RL) is a type of machine learning algorithms where a policy is searched by an agent to solve sequential decision tasks. In RL, the policy learns the desired behavior to maximize a total expected reward through interaction with a dynamic environment by trial and error (Du & Swamy, 2019). Deep Reinforcement Learning (DRL), integrates neural networks to reinforcement learning. In the last years, the most common approach for reinforcement learning applications in robotics was deep neural networks. In this line of research, multiple DRL works have been proposed for industrial robot manipulation (Haarnoja et al., 2018; Levine et al., 2016; Mnih et al., 2015).

In comparison to general imitation learning approaches, RL usually requires significantly more examples to train effective policies (Ravichandar et al., 2020; Sun et al., 2017). Empirical results have shown that training over IL requires fewer demonstrations than RL, in other words, IL is more sample efficient than RL (Abbeel et al., 2010; Kober & Peters, 2009; Osa, Pajarinen, et al., 2018; Shin et al., 2019). With a growing number of possible states and actions in RL, the data and computational requirements increase rapidly and can become infeasible when the number of features grows even for problems with discrete states (Yueyue Liu et al., 2020). To overcome the sample efficiency problem in RL, (B. Kim et al., 2013; Nair et al., 2018; Richter et al., 2019; Taylor et al., 2011; Vecerik et al., 2018; Zhu et al., 2018) have combined reinforcement learning with imitation learning to reduce the amount of exploration required.

Applying RL approaches to robotics imposes several challenges. RL methods require reward functions that are well-shaped and carefully designed. Such reward functions are difficult to specify in-advance (YuXuan Liu et al., 2018). Learning policies with RL can be challenging for robotic problems with high-dimensional continuous states and actions (Kroemer et al., 2019).

Moreover, the reward functions are specifically designed for each task (Hussein et al., 2017) which can limit their application to a wide variety of manipulation tasks.

In contrast, a neural network policy learned using deep imitation learning can be extended to new tasks with minimal changes in the framework. This can be achieved by applying the same learning architecture to individual datasets for each task (Rahmatizadeh et al., 2016a; T. Zhang et al., 2018) or by learning different manipulation tasks jointly (R. Rahmatizadeh et al., 2018). Thus, deep imitation learning can be formulated for a broad range of manipulation tasks without the need to design specific reward functions. Furthermore, imitation learning can be particularly relevant for tasks where it is desired to not only accomplish the goal of the task but to also mimic the behavior of the human demonstrations. For example, an autonomous surgical assistant could better collaborate with a surgeon if the robot motions resemble what the surgeon would naturally expect from a human assistant.

2.4 Surrogates for surgical robots

Developments in medical robotics span several research areas and disciplines, from hardware enhancements (e.g. tools design, flexible robotics) to software developments (e.g. trajectory optimization, machine vision). Key aspects currently being studied by ergonomists, psychologists, and social scientists, in general, include multimodal communication, effective teaming, cognitive load, and interface usability assessment. These studies are usually done using virtual simulation or real-robotic systems. While the first option is more accessible, it provides lower face validity. Conversely, access to real medical robots imposes a challenge for most students given that they are scarcely available in teaching hospitals. When accessible, the studies are done using RAS research platforms available only to a limited number of research institutions.

Two prominent RAS platforms are currently available to the research community: The da Vinci Research Kit (dVRK) from Surgical Intuitive and the RAVEN II research platform (Takacs et al., 2015). The dVRK is a research kit consisting of a collection of robotic modules from the first-generation da Vinci Surgical System (*Da Vinci Surgery*, 2020). The platform is used to perform computer-assisted surgery for both autonomous and teleoperated robots. The RAVEN II is an especially-aimed open platform for research in surgical robotics, and has been used to explore robotic planning in surgical tasks and teleoperation (Hu et al., 2015; Ben Kehoe et al., 2014). RAS

platforms such as MiroSurge and OP: Sense have also been developed as non-open source and with limited availability. Miro-Surge, developed by the German Aerospace Center, consists of three 7-degrees-of-freedom (DOF) robotic arms and specialized instruments that include force sensors for haptic feedback (Tobergte, 2010). OP: Sense is a research platform for surgical robotics developed by the Karlsruhe Institute of Technology that encompasses two KUKA LWR IV arms and a Stubli RX90 robot (Mönnich et al., 2012). SRI has made available the 7-DOF Taurus Robot, which was used to explore innovative HRI modalities (T. Zhou et al., 2016). More recently, a research platform has been proposed for micro robot surgery training (D. Zhang et al., 2020).

Although these platforms allow the research community to gain access to medical robots and high-level control routines, the cost of these technologies is still prohibitive for the majority of scientists and students interested in "hands-on" experience. For example, the dVRK is freely available, but controllers are required to operate the robot at a cost of over $20K assuming the availability of an experienced technician in order to install the system. In contrast, the cost of industrial robotic systems is only a fraction of the cost of the previously mentioned surgical robots. While all industrial robots may not fulfill all human-robot safety requirements or possess the necessary kinematics for surgery procedures, they still can be useful to practice the fundamentals of RAS (Shademan et al., 2016).

2.4.1 Research platforms for RAS

The dVRK is a complete platform for performing, programming, and simulating telerobotic mock surgical procedures. The system can be controlled with open-source software made available by (Kazanzides et al., 2014) using the Robot Operating System (ROS). The hardware can be obtained from Intuitive Surgical and consist of two master tool manipulators, two manipulators on the patient side, a stereo display, and a set of foot pedals. The da Vinci Research Kit community includes 30+ institutes and universities (Bolzoni Villaret et al., 2017).

The Raven II robot was initially developed as an open-source platform for research at the University of Washington (Li et al., 2019). This system has two 3-DOF positioning system that can attach 4-DOF laparoscopic tools and one removable gripper. The software is built on Linux and ROS to facilitate software development between several universities and the robot research community.

The MiroSurge is a telepresence system for Minimally Invasive Surgery (MIS) that allows research on advanced user interaction modalities. The system is composed of 2 compliant MIRO robotic arms, 2 specialized instruments (MICA) for laparoscopic and open surgery, and 2 haptic input devices. The MICA tool assembled in the MIRO arm provides 6-DOF of motion control and force/torque measurements in all DOF (Tobergte, 2010).

The microsurgical robot research platform (MRRP) is a robotic system developed for research in robot-assisted microsurgery (D. Zhang et al., 2020). The system has two 6-DOF robot arms controlled using a haptic input device and a hand-held custom controller. Two microscope, and multiple exchangeable microtools such as needles and forceps are included in the system.

The MAZOR X system is a recently released robot for spinal surgery that comprises a workstation and a mechanical surgical arm (D'Souza et al., 2019). MAZOR X is a commercially available robot intended to provide planning and guidance during spine surgical procedures.

The University of Nebraska-Lincoln has developed a series of single-incision robots designed for diverse surgical procedures (Dolph et al., 2019). The robotic systems include a base attached to two miniaturized robot arms designed to maneuver completely inside the peritoneal cavity to perform manipulation and cautery. The arms can fold inwards to allow for the insertion of the device in the peritoneum cavity through a single incision.

The SmartArm system is a recently developed surgical robotic research platform for constrained spaces (Marinho et al., 2020). The system is comprised of two industrial robot arms, each equipped with actuated flexible instruments such as forceps. The system is integrated with Linux and the robot operating system. The previous platforms can be costly, and often require a license agreement with robot manufacturing companies.

Nowadays, industrial robots are highly effective and popular for repetitive and labor-intensive tasks due to their positioning accuracy, repeatability, durability, and robustness. One example of the use of industrial robots in surgery is the positioning of laparoscopes and FDA approved tools. Shademan et al. (2016) developed a surgical system to perform soft tissue surgery through supervised autonomy employing a KUKA robot. The robotic platform displayed positioning repeatability within 0.5mm. Capolei et al. (2017) proposed an autonomous robotic solution based

on an industrial manipulator to control the position of a laparoscopic camera. In Bolzoni Villaret et al. (2017) a hybrid robotic system was developed to assist in endoscopic skull base surgery. (X. Zhang et al., 2020) proposed the use of a UR5 industrial robot for control of a flexible tendon driven endoscope.

2.4.2 3D Printing in surgical robotics

Three-dimensional (3D) printing allows a quick and low-cost transition between conceptual design and a fully functional device optimized through iterative prototyping cycles. There are currently a variety of techniques integrating 3D printed parts with diverse tools in surgical systems. (Y. Kim et al., 2017) designed a 3D printed robot developed for brain tumor removal. Rateni et al. (2015) developed a grasping tool for MIS, based on a soft gripper created with an elastomeric material. The tool was adapted to the end-effector of a 6-DOF robotic platform using a rigid 3D printed base. Chandrasekaran et al. (2017) presented the design of an expendable tooltip for robotic surgery printed on a photopolymer. They propose a solution for power transmission based on magnetic coupling. (Guo et al., 2018) designed a 3-D printing disposable transmission system for a surgical robot to manipulate non-magnetic tools. Gerboni et al. (2016), developed a surgical gripper that combines metallic tooltips, a 3D printed mount, and linear actuators made of elastomeric material. These works show the feasibility of manufacturing processes for the development of tools for robotic surgery.

Rigid plastic material has been used as the main material of 3D printed tools for surgical grippers. McKinley et al. (2016) proposed a system with interchangeable surgical instruments. The system is composed of 3D printed parts, namely a mounting mechanism for the da Vinci Surgical System, a tool-guide, and a sleeve to autonomously perform the change of surgical instruments. (Sheng et al., 2018) proposed a flexible robotic catheter developed with 3D printing technologies designed to ablate heart tissue. Mintenbeck et al. (2014) designed a 3D printed cable-driven snake robotic instrument with 4-DOF for MIS. The instrument was designed to be operated with the industrial robot KUKA LWR IV. (D. Zhang et al., 2020) has proposed the use of 3-D printing connectors for mounting microtools on the MRRP robotic platform. The previous systems leverage 3D printing technologies and fast prototyping to explore designs compatible with the current surgical robotics standards.

2.5 Surgical tasks

Autonomy and/or semi-autonomy can release the operator from performing tasks that can easily be automated. Incorporating semi-automation in the operating room has been suggested to alleviate the cognitive load experienced by surgeons as well as to enable telesurgery in remote locations where there is limited medical personnel (Garcia et al., 2009).

2.5.1 Skill and task learning in surgical robotics

Task decomposition of complex surgical procedures into primitive motions or subtasks can help in task analysis and the automation of such tasks. A complex surgical task can be learned by dividing the task into a set of meaningful motion sequences that are learned independently. This process is known as surgical skills modeling (Moustris et al., 2011). These motions form a library of skills that can be learned and reused to execute high-level surgical tasks. An example of a skill that can be reused is the "pull" motion, which can be used for different tasks such as suturing and knot-tying. In surgical skills modeling, each task is comprised of a series of atomic elements referred to as primitive actions or *surgemes* (Gao et al., 2014). For instance, a suturing task is composed of the surgemes "reach for the needle", "position needle", "insert needle" and "pull/push the needle" etc.

Surgemes have been used to model surgical gestures, and for motion generation in surgical task automation (Krishnan et al., 2015; Nagy & Haidegger, 2019; Reiley et al., 2010), surgical skills assessment (Reiley & Hager, 2009; Varadarajan et al., 2009) and in the recognition and segmentation of surgical actions (Béjar Haro et al., 2012; Lin et al., 2006; Madapana et al., 2019).

Early approaches in surgical automation have focused on generating robot trajectories for automating all or some of the primitive actions that compose a task. Mayer et al. (Mayer et al., 2006) have employed recurrent neural networks to predict the trajectory of the winding portion of the knot-trying task, while the remainder of the task is preprogrammed. Berg et al. (2010) proposed a framework that utilizes Dynamic Time Warping (DTW) for temporal alignment of demonstrations to create smooth reference trajectories for a two-handed knot tie. Similarly, Reiley et al. (2010) modeled independent surgemes by using a gaussian mixture regression to extract a

smooth reference trajectory. However, these approaches focus on modeling individual sections of the task and assume a predefined order of primitive action execution.

Later works evaluate the ability to generalize the demonstrated trajectories to variable settings during testing. (Schulman et al., 2013) presents an approach that finds a wrapping function that directly maps the demonstrations of a simplified suturing scenario into a test scene. The method cannot generalize to high variations in the environment, as the performance of the task gradually decreases for higher perturbations of the initial conditions. In contrast, (Osa, Sugita, et al., 2018) presented an approach that completed an online trajectory planning of a knot-tying task under changing conditions of the environment successfully. Their work used DTW to remove the temporal variance of the trajectory demonstrations and used Gaussian process regression to model the trajectory distribution given the current state of the system. These approaches focus on modifying demonstrated trajectories to adapt to new environment conditions.

Alternative approaches focus on generating algorithms that automate a task by coordinating predefined or learned primitive actions. Approaches that combine primitive actions to perform complex tasks are referred to as high-level learning (Chernova & Thomaz, 2014). (Padoy & Hager, 2011) proposed a high-level approach to combine manual and automatic subtasks in a suturing scenario. Their work presented a learning from demonstration approach that used Hidden Markov Models (HMM) for recognition of subtask completion. Once the subtasks were recognized, the execution of specific task segments was triggered based on a predefined order of motion sequence execution. In such approach, DTW was used to learn manually segmented primitives of a suturing task. Another high-level approach based on learning by demonstration was presented by (Murali et al., 2015) for the debridement and pattern cutting tasks. That approach relies on motion sequences that are identified and parametrized by experts. These motion sequences and the state of the environment obtained from a stereo camera were used to construct a Finite State Machine (FSM) to execute each task. In that work, empirical parameters were established in order to constrain the execution of motion sequences. For example, in the debridement task, once the target piece was grasped, it was retracted by a predetermined height. In addition, such approaches for high-level learning require expert knowledge to carefully design the finite states that compose the controller as well as to model the transitions between states.

Automation approaches that model primitive actions can rely on the manual identification and segmentation of the demonstrations. To address this limitation, approaches for autonomous task segmentation have been proposed to alleviate the labor of manual segmentation (Fard et al., 2017). However, generating autonomous task execution algorithms for complex tasks is complicated because: (1) we need to segment the task into meaningful primitive actions, (2) the automation of each primitive action and the design of high-level controllers are required to complete the whole task, (3) we require to assign labels to highly variable time-series data (Yip & Das, 2017).

Moreover, the process of formulating adequate models and the definition of its parameters may consume significant time (Kassahun et al., 2016). For example, FSM models require the specification of meaningful states in the task as well as feasible state transitions. Learning individual models for a primitive action requires the segmentation of multiple trajectory demonstrations. This process can be particularly limiting when such steps need to be repeated for each individual task. In contrast, deep learning approaches are appealing because they can allow learning implicit models directly from sensory data.

2.5.2 Deep learning in surgical tasks automation

Deep learning has been recently proposed in combination with reinforcement learning for autonomous manipulation of soft tissue, which is a task frequently preformed in surgical procedures (Shin et al., 2019). In particular, Deep Reinforcement Learning (DRL) has been proposed for controlling tensioning policies of a tissue phantom while scissors execute predefined contour trajectories in a pattern cutting task (Thananjeyan et al., 2017). Their work proposed DRL to learn tensioning policies of one pinch point during a 2D pattern cutting task using a finite-element simulator. Their approach was extended in later work to learn tensioning directions of multiple pinch points over more complex contours (Nguyen et al., 2019). Such algorithms determine the direction of tension using the current state of the phantom sheet, the predefined trajectory, and the position of the cutting tool. In such approaches the environment state uses a model (mesh points on the tissue phantom) that is specific for the tensioning task and could not be directly applied to general surgical manipulation tasks.

A different common surgical task is debridement. Debridement consists in the removal of foreign matter from a wound (Steed, 2004). In this context, a reinforcement learning algorithm has been

proposed to solve simple tasks in simulation by learning individual policies for the action primitives "pick" and "reach" that compose the task for one arm manipulator (Richter et al., 2019). Their work demonstrated a debridement task as a sequence of two actions, "pick" to pick the part from a known goal location and "reach" for handling the part to an assistant. RL was used for automating primitive actions in a mock-up surgical setting where each primitive action had a corresponding predefined goal. However, the approach has not yet demonstrated to be successful for tasks that require changes in tool-reorientation, as they simplified the problem by assuming the tool orientation to be fixed. Their work exploited the benefits of behavioral cloning for learning the "pick" policy as they were not able to learn the policy using RL alone.

Reinforcement learning has been used in combination with imitation learning for autonomous tissue manipulation using two robot arms. (Shin et al., 2019) compared the performance of 2D arm motion policies obtained through reinforcement learning and imitation learning. Their work focused on learning motion policies for two robot arms in the 2D plane. The goal was to align points on a tissue with a desired position in the image. The authors determined that when the demonstrations encompass a wide range of desired positions, imitation learning was sufficient and additional exploration using RL was not necessary.

A difficulty associated with reinforcement learning is that each additional feature added to the state increases the size of the solution space exponentially (Burton, 2010). For example, including tooltip orientation as part of the robot state would increase the dimensions of robot solution space (e.g. modeling orientation in 3 directions would add 3 additional features to the robot state). Policy learning can be challenging for RL in high-dimensional spaces (Kroemer et al., 2019). For this reason, additional research is required to explore the applicability of RL approaches to surgical tasks with high dimensional spaces. Examples of this, are surgical tasks with bimanual manipulation as well as end-effector position and orientation control.

Deep Imitation Learning has been proposed as a feasible method for full tasks learning from demonstrations. (J. W. Kim et al., 2019) have proposed a DIL approach for autonomous surgical tools navigation inside the eye. In that work, a deep network was trained to imitate trajectories that were artificially generated based on information on the eye geometry. Data augmentation was achieved by varying the starting position of the trajectories. Although the approach simplified the

variability of the task by modeling artificial trajectories as straight lines and keeping the eye position and the tool orientation fixed, it demonstrated the feasibility of DIL methods.

In summary, several methods have been recently introduced that show the potential of RL and, in general, IL for automation of surgical subtasks. These works have primarily modeled sections of the task and have limited the exploration space of the policy by keeping the tool orientation fixed or by modeling only one arm. Further research is desired to present deep imitation learning approaches that can be used for more complex surgical tasks such as peg transfer and knot-tying. These tasks require to model orientation for successful completion of the task.

Deep Imitation Learning approaches have the potential to learn full multi-step tasks without the need for previous task segmentation. Furthermore, in DIL, it is not necessary to specify reward functions and goals for each primitive action. This would be challenging and onerous for multi-step surgical tasks and would require expert domain knowledge.

2.5.3 Surgical tasks

The peg transfer task

The peg transfer task involves the transfer of parts between pegs on a board, using the two robotic arms, through teleoperation. The task consists of picking the part with one gripper, pick it up, and transfer to the second arm, which is responsible for depositing it on top of a different peg. Due to the dexterity and precision required to solve this task, it is a key skill included in the Fundamentals of Laparoscopic Surgery (FLS) for surgical skills training (Ritter & Scott, 2007). The peg transfer task has attracted interest from the surgical robotics community for the evaluation of robot teleoperation interfaces (Abiri et al., 2019) and measurement of surgeon skills (Bark et al., 2012; Brown et al., 2017; Peng et al., 2019). This task can be evaluated at the trial level, where one transfer from peg to peg is assessed, or at the surgeme level (Madapana et al., 2019; Rahman et al., 2019). As opposed to (Madapana et al., 2019; Rahman et al., 2019) in this thesis the goal is to learn to perform the whole task autonomously at once rather than a sequence of connected surgemes.

In the robotics automation community, the peg transfer task has been identified as a challenging task to assess autonomous execution due to the dexterity required and the limited clearance between the blocks and pegs (B. Kehoe et al., 2014). Next, we describe a few attempts that have

been made for the automation of the peg transfer task using only one arm (no handover). (Rosen & Ma, 2015) proposed the use of finite state machines for path planning and task planning to automate the handover-free peg transfer task. The approach sequentially repeats steps 1 to 8 until the final state of the task has been reached. That approach employs artificially generated trajectories defined offline for each state as well as specific speed patters for each state transition. Another approach for automation of the handover-free version of the peg transfer task was proposed by (Hwang et al., 2020). In their work, they programmatically define each motion sequence based on detected candidate grasp points and executed the sequences in a predefined order.

The aforementioned approaches for peg transfer automation do not account for the interaction between the two robotic arms during same object manipulation, which is required to achieve the original peg transfer task according to the FLS. (Nagy & Haidegger, 2019) proposed the automation of the dual-arm peg transfer task by using a hierarchical motion planning that executed a given sequence of independent surgemes. Each surgeme is executed based on a set of parametrized actions, obtained from the environment. For example, "grasp" consists of navigating to the parametrized approach position, then navigating to the grasp position, and finally executing the action to close the jaws. The previous approaches for peg transfer automation rely on a set of actions or motion sequences that were identified for each section of the task. Furthermore, such approaches require the user to program primitive actions and specify action parameters for successful task execution. In our work, we aim at learning a continuous controller for task execution from human demonstrations without explicit models for individual primitive actions.

The surgical debridement task

The surgical debridement task consists of removing diseased tissue from the neighboring healthy tissue. It is considered beneficial to the treatment of conditions (Granick et al., 2006) such as necrotic tissue (Steed, 2004), burn injuries (Duteille & Perrot, 2012), and non-healing wounds (J. Liu et al., 2011). (Seita et al., 2018) has proposed an approach for autonomous debridement that focused on developing a calibration procedure for cable-driven surgical systems such as the dVRK. In that work, the goal involved approaching detached foreign objects on a planar surface, grip them,

and transfer them to a container. That work relies on specific task handcrafted features such as approaching the target object with a fixed orientation.

In a similar setup, (B. Kehoe et al., 2014) has demonstrated autonomous surgical debridement by using state machines for specifying high-level task integrated with a motion planning algorithm for preforming low-level subtasks. In that work, the surgical debridement consisted of retrieving fragments placed randomly on a planar work surface. In their system, the two arms did not interact with each other but could operate simultaneously to perform the task independently. The low-level motion planning algorithm plans optimal, collision-free trajectories for both arms while imposing collision, pose, and feasibility constraints. In a simulation environment, (Richter et al., 2019) implemented an RL method for the automation of the debridement task. In their work, the debridement task was segmented in two actions "pick" to collect the sample and "reach" to move the sample to a disposal location.

Alternatively, the debridement task can also be designed to make use of the two arms which is the setting commonly used in surgical robotics. One arm is used for grasping the damaged tissue, and the other for separating it from the healthy tissue. (Nichols et al., 2015) implemented a finite state machine (FSM) to solve the dual-arm debridement task under different levels of autonomy. That work defined robotic actions such as "grasp", "ungrasp" and "cut" which are sequenced to solve the autonomous debridement task. (Murali et al., 2015) proposed an FSM for automation of surgical debridement based on human demonstrations to retract and cut fragments embedded in a viscoelastic material that represent damaged tissue. In their approach, they identify, segment, and parametrize sequence of motions and sensorial information to construct a finite state machine for each task. Those works require to define the specific states that constitute the debridement task as well as the actions the robot can take at each step. Furthermore, such approaches require the definition of task-specific parameters which are generally determined empirically such as the depth of tissue retraction, gripper actions (open/close), the direction of motion for cutting, etc.

In our work, we consider the debridement task from grasping the fragments, retracting, cutting the fragments to separate from the remaining tissue, and moving the debrided fragments to a surgical tray. We trained a continuous controller from human demonstrations to map robot and

environment states to actions of the robot without explicit definition of surgeme intrinsic parameters, nor the task primitives.

2.6 Deep imitation learning for robot manipulation

Recent approaches in deep learning have demonstrated the capability to model full manipulation tasks by learning from demonstrations. Convolutional Neural Networks (CNN) have been proposed to learn robot control policies for manipulation tasks. This includes end-to-end policies that map raw images and robot configurations to output motor torques (Levine et al., 2016). (T. Zhang et al., 2018) demonstrated the applicability of CNN policy representations for DIL in a variety of manipulation tasks. The latter approach used convolutional layers to process raw images and robot configurations from the last five steps to predict the linear and angular velocity of one robot manipulator as well as the gripper state. Their work demonstrated that neural networks are a feasible for policy representation in deep imitation learning using behavioral cloning. Such approaches indicate that the natural variations in human demonstrations allow better generalization in the imitation learning policy.

As manipulation tasks can be represented as temporal sequences, Recurrent Neural Networks (RNN) (Mayer et al., 2006) have been proposed as an alternative to model time dependencies in robot execution. A class of RNN's that models long time dependencies, is the Long-Short Term Memory Networks (LSTM) (Hochreiter & Schmidhuber, 1997). LSTM's have been proposed to train visuomotor controllers in DIL for robot manipulation that output the robot's joint control parameters based on the robot configuration and a third-person view image of the robot and the environment. James et al. (James et al., 2017a) have proposed a CNN+LSTM method to transfer visuomotor control policies trained from artificially demonstrations into a real system for a pick and place task. In that work, abundant demonstrations (~4000) were collected by moving the robot's position over linear trajectories between key points in the task using a simulation environment. Examples of such key points are a point directly above the target object for executing the grasping action and a point located above the basket where the object needed to be placed.

Another DIL visuomotor policy based on LSTM+CNN was proposed by (Rahmatizadeh et al., 2018) to predict joint configuration commands from a third-person view image. The input image includes a view of the robot and the objects of interest. In this approach, the trajectory

demonstrations were obtained from a motion controller or by tracking the motion of the user's hand. The visuomotor policy is tested for tasks such as push a tool to a given pose. These approaches demonstrate the ability of LSTM networks to represent deep imitation learning policies. Such policies aim to reproduce a desired end-effector's position from human demonstrations or artificial trajectories. The previous DIL works predicted robot joint states as the output command and did not explicitly model robot end-effector orientation as part of their state-action space.

Alternatively, a deep imitation learning approach can explicitly account for robot orientation by encoding the robot state and action based on the robot pose and gripper state (open/close). This state-action representation has been proposed by (Rahmatizadeh et al., 2016a) to train LSTM networks policies for simple manipulation tasks such as pick and place of a cube and push cube to pose. They showed that recurrent networks (RNN), particularly LSTM, perform better in such manipulation tasks than direct feedforward neural networks due to their capability to encode long temporal dependencies. However, their application did not explicitly require precise end-effector control and additional research is required to determine whether such representation is sufficient to learn policies for tasks that are sensitive to accurate robot pose control.

As surgical tasks often involve the collaboration of two robotic arms such as the peg transfer and the knot-tying tasks, there is a need to study DIL algorithms that can model such interaction between two arms (or more). In previous work, this interaction was modeled by learning the behavior of each arm independently and reproducing their actions sequentially. For example, (Padoy & Hager, 2011) have modeled the suturing scenario as a predefined sequence of actions of independent end-effectors such as "grasp needle with right arm" or "pull thread out using left arm". More recently, (Murali et al., 2015) have attempted to learn such interactions from segmented sequences of the data, by modeling the primitive actions for a cutting task as a multi-arm maneuver. Approaches that are used to learn robot arm interactions from demonstrations can be relevant to learn user preference in task execution. For instance, in tasks such as the peg transfer, the user can naturally prefer one of the robot arms for picking up an object considering the direction of the peg transfer (e.g. left to right) and would use the opposite arm to place the object. It is of interest to generate task controllers that can model the interaction between different end-effectors according to user preferences, as their preferences can change with task conditions.

Dual-arm manipulation task coordination is not only essential in surgical robots, but there is also an increasing interest in anthropomorphic robotics and bimanual industrial manipulation (Smith et al., 2012). Some approaches model the coordination of two robotic arms explicitly, by detecting temporal dependencies using HMM's (Asfour et al., 2006). (Gribovskaya & Billard, 2008) relied on a dynamical systems based approach to encode coordinated motion patterns. (Pairet et al., 2019) modeled dual manipulation as a closed-chain system. (Pavlichenko et al., 2018; Tamei et al., 2011; Yuba et al., 2017) modeled the coordination of two robotic arms implicitly.

In our work, we propose to exploit LSTM for training policies that jointly optimize for task automation while successfully addressing two-arm coordination. This is in contrast to modeling each subtask and robotic arm independently. We also evaluate the pose control policy for tasks that rely on accurate orientation control of the end-effector's and we consider the generalization of the policy for high variations in unseen testing settings.

2.7 Long-Short Term Memory (LSTM) networks

Recurrent Neural Networks (RNN) are a type of Artificial Neural Networks (ANNs) that can consider previous inputs and states and use that information to predict the outputs by exploiting recurrent connections in the hidden layers (Diamantaras et al., 2010). In addition, LSTMs are capable of learning long-term dependencies in sequential data (Hochreiter & Schmidhuber, 1997). LSTMs have been employed in numerous robotic applications, such as motion planning (Finn & Levine, 2016; Inoue et al., 2019), manipulation (Mott, 2019; R. Rahmatizadeh et al., 2018, p.; T. Wang et al., 2019) and human-robot interaction (Alahi et al., 2016; Y. Wang et al., 2017). In the medical field, it has been used to generate trajectories for the winding portion of the knot-tying task (Mayer et al., 2006), recognizing surgical activities (DiPietro et al., 2016) and surgical skill evaluation (Ezziyyani, 2020)

Stacking LSTM layers and generating direct connections between non-consecutive layers have been proposed to model deep LSTM networks as well as to avoid vanishing gradients (Gui et al., 2019; Pundak & Sainath, 2017; Sak et al., 2014). A stacked LSTM architecture is a model composed of several LSTM layers representing multiple levels of abstractions over time (Graves et al., 2013). This makes stacked LSTMs a stable technique for challenging sequence prediction

problems (Brownlee, 2017), such as robot action prediction for manipulation tasks. An illustration of a stacked LSTM with skip-connections is shown in Figure 1.

Skip-connections are direct connections between non-consecutive layers. Such connections have been found to facilitate training of deep LSTMs (Graves, 2014) by reducing the number of processing steps between different layers of the network. In robot manipulation, skip-connections have been previously employed to facilitate learning of LSTM of robot control policies in DIL (Rahmatizadeh et al., 2016a; R. Rahmatizadeh et al., 2018).

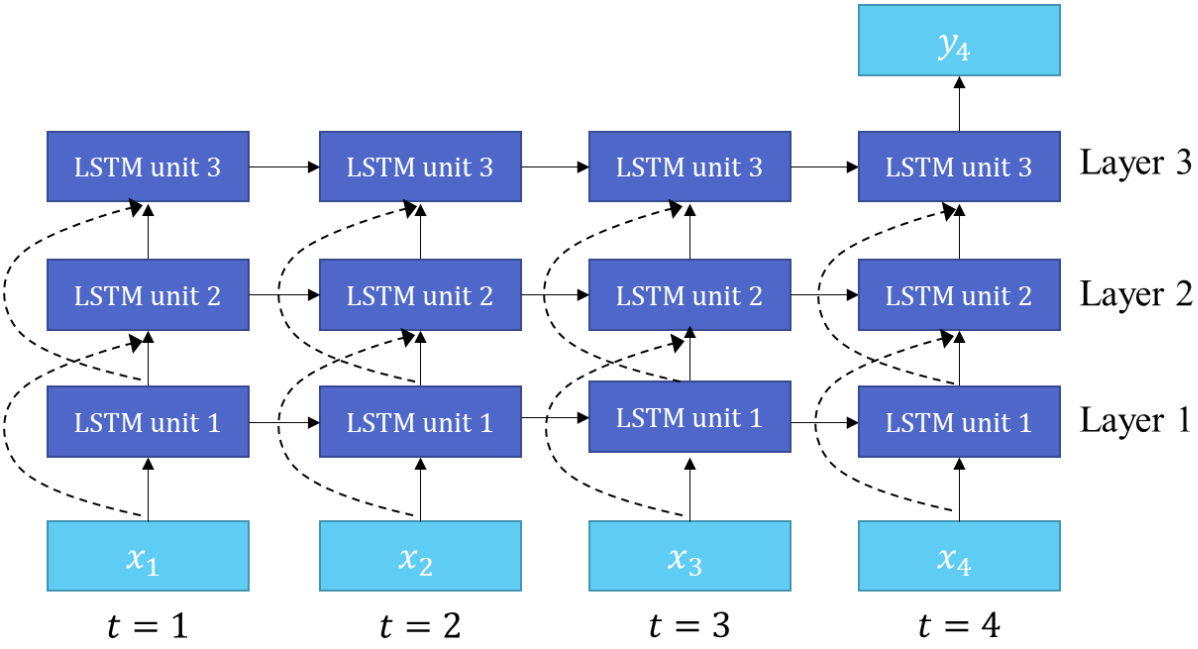

Figure 1. Stacked LSTM of three layers for 4 time steps with skip connections between the inputs *x* and the layers 2 and 3. In this example, the output of the LSTM network is y, which is predicted using the sequence of inputs *(x1,x2,x3,x4)*

In DIL policy learning, the inputs of the network (*x)* can be, for example, the state of the system encoded as the pose of the robot manipulator in the coordinate space. The output of the LSTM network (*y)* are the actions the system can take, which can encode for example the next end-effector joint states. For the example shown in Figure 1, the LSTM network uses the state of the previous 4 time steps *(x1,x2,x3,x4)* to predict the action *y4* at time t=4.

2.8 Quantitative metrics in surgical robotics

The cumulative sum (CUSUM) analysis is a technique adopted by professionals in the medical field to analyze the learning curve of surgical procedures (Bokhari et al., 2011). The CUSUM technique has been applied for the analysis of single surgeon learning curves in several surgical fields such as thyroidectomy (Liao et al., 2014), colorectal surgery (Bokhari et al., 2011), rectal cancer (Park et al., 2014; Yamaguchi et al., 2015), and gastric endoscopy (Yoshida et al., 2017).

A surgeon learning curve (Khan et al., 2014) can be represented, for example, as the procedure completion time versus the number of procedures completed. The CUSUM analysis consists of transforming the raw data of the curve (e.g. completion time) into the cumulative sum of the data deviations from their mean. The cumulative sum can then be plotted to enable visualizing data trends, which are used to determine different learning phases of the curve. Such phases are used as an indication of different levels of surgeon expertise.

Quantitative metrics have been proposed to measure surgical skills in open surgery and minimally invasive surgery. Path length is a widely used metric for skill evaluation that measures the length of the trajectory followed by the tip of the instruments during the task. Several researchers have found path length to be an objective metric to measure surgical skills (Clinkard et al., 2015; Rivard et al., 2014; Vedula et al., 2016). Task completion time is another common metric for skill assessment, which has been previously used to distinguish different levels of expertise (Hofstad et al., 2013; Oropesa et al., 2013; Reiley et al., 2011). Similar to path length, rotational changes in tooltip orientation during procedures can be used as time-invariant metrics to evaluate surgical skills. (Hofstad et al., 2013; Kundhal & Grantcharov, 2008) have evaluated angular length metrics by measuring the total change in instrument orientation throughout the procedure. The previous metrics will be used to make quantitative comparisons between robot generated trajectories and human demonstrations.

2.9 Discussion

The field of surgical robotics can benefit from controllers that merge dual-arm coordination with complete tasks modeling. Deep learning allows efficiently learning state-action controllers for tasks relying on two (or more) interacting robotic arms. Deep imitation learning approaches have

demonstrated to be an attractive venue for modeling policies for multiple manipulation tasks from demonstrations. Moreover, the policies obtained using DIL resemble human teacher behavior without the need for carefully defined reward functions. Furthermore, DIL frameworks can be directly applied to diverse execution tasks by training the network over different datasets.

2.10 Summary

This section discussed imitation learning in robotics, with an emphasis on applications of robot manipulation and, in particular, surgical robotics. Additionally, industrial robotics and 3D printing technologies are discussed in the context of surgical robotics platforms for robotic-assisted surgery. Then, a review of prominent task execution approaches in surgical robotics is presented. Then, an overview of deep imitation learning for robotic manipulation is discussed as well as their applicability to surgical robotics. Finally, task assessment metrics are discussed in the context of human vs. robot task execution.

3. METHODOLOGY

This section describes the main methods used for the formulation of an imitation learning approach for learning dual-arm dexterous manipulation tasks using long-short term memory networks (LSTMs). The main components of the robot learning approach are presented in Figure 2. The imitation learning algorithm is based on a three-layer stacked LSTM network with skip-connections that learns a robot control policy for task execution. The learned policy π_θ uses as input the current observation o_t to predict the robot arm's poses (action) $a_t = \pi_\theta(o_t)$ that will be sent as commands to the robotic system. The current observation $o_t = \{ s_{(t-n:t)}, e_{(t-n:t)} \}$ is composed of the robot state $s_{(t-n:t)}$ and the state of the environment $e_{(t-n:t)}$ from the previous n time steps. The network is trained based on demonstrations from robot teleoperation collected by one user in a virtual reality setting.

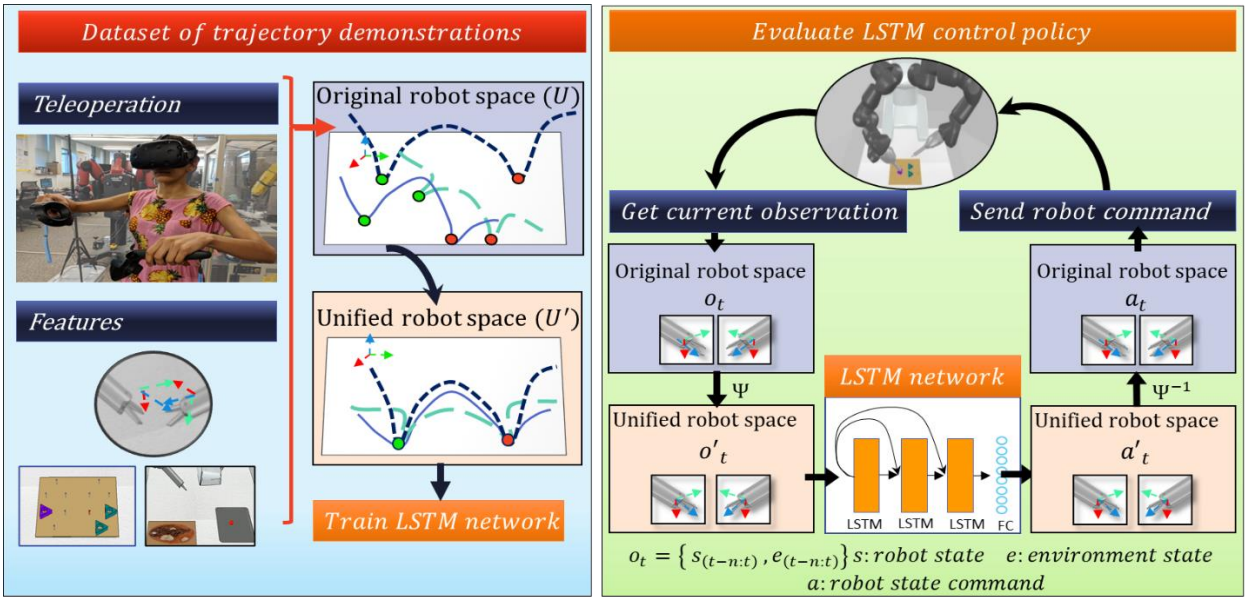

Figure 2. Overview of the approach for robot task learning of dual-arm tasks.

The original robot coordinate space in which the demonstrations are collected is denoted with U. An affine transformation Ψ is applied to the demonstrations in the dataset $D_{task} = \{(o_t^{(i)}, a_t^{(i)})\} \in U$ (set of observations o_t and corresponding actions a_t) to map them to a robot coordinate space U' where the start position of the object of interest X_s and its goal position X_g are aligned.

$$\Psi\colon \tau_i = \{o_i, a_i\} \in U \rightarrow \tau'_i = \{o'_i, a'_i\} \in U' \quad (1)$$

This coordinate space representation will be referred to as the "unified robot coordinate space" denoted with U', where $\Psi\colon U \longrightarrow U'$. The dataset represented in the unified robot coordinate space is denoted as D'_{task}. The policy π_θ is represented by an LSTM network trained with the dataset represented in the unified robot space, which allows to learn a policy for the execution of complex tasks from demonstrations provided by a human teacher.

During robot execution, the affine mapping Ψ is used to transform the current observation o_t to the unified robot space U' where the LSTM network is evaluated.

$$\Psi\colon o_t = \{s_{(t-n:t)}, e_{(t-n:t)}\} \in U \rightarrow o'_t = \{s'_{(t-n:t)}, e'_{(t-n:t)}\} \in U' \quad (2)$$

The policy π_θ utilizes the observation in the unified robot space to predict the next robot action.

$$a'_t = \pi_\theta(o'_t) \quad (3)$$

The robot action $a_t' \in U'$ predicted by the network policy is mapped back to the original robot space U and sent as the next control command a_t to the robotic system.

$$\Psi^{-1}\colon a'_t \rightarrow a_t \quad (4)$$

To validate the method in an online simulated setting, two networks have been trained to execute relevant mock-up surgical tasks (one network for each task) that require the coordination of two robotic arms: the debridement and peg transfer tasks.

3.1 The robotic system

The robotic platform used for this work is the industrial collaborative robot from ABB, YuMi which has been adapted as a surgical research platform using rapid prototyping. In this section, an approach to adapt the industrial robot as a surgical research platform is presented. 3D printed gripper adaptations have been designed to transform the robot end-effector's parallel motion into a gripping motion. The goal of this was to allow the tip to operate different tools for mock-up surgical tasks. Moreover, a simulation environment was developed for the adapted YuMi robot to be used as a platform for acquiring human demonstrations and as the testing platform for autonomous task execution (Sanchez-Tamayo & Wachs, 2018).

3.1.1 Gripper adaptations for surgical tasks

The industrial robot

The ABB YuMi Robot is a collaborative robot with a pair of 7-DOF arms (Taghbalout et al., 2019). Each arm is equipped with a parallel-jaw gripper and two pneumatic modules that can carry a payload of 500 grams with an accuracy of 0.02 mm (ABB, 2019) and gripping force of 20 Newtons.

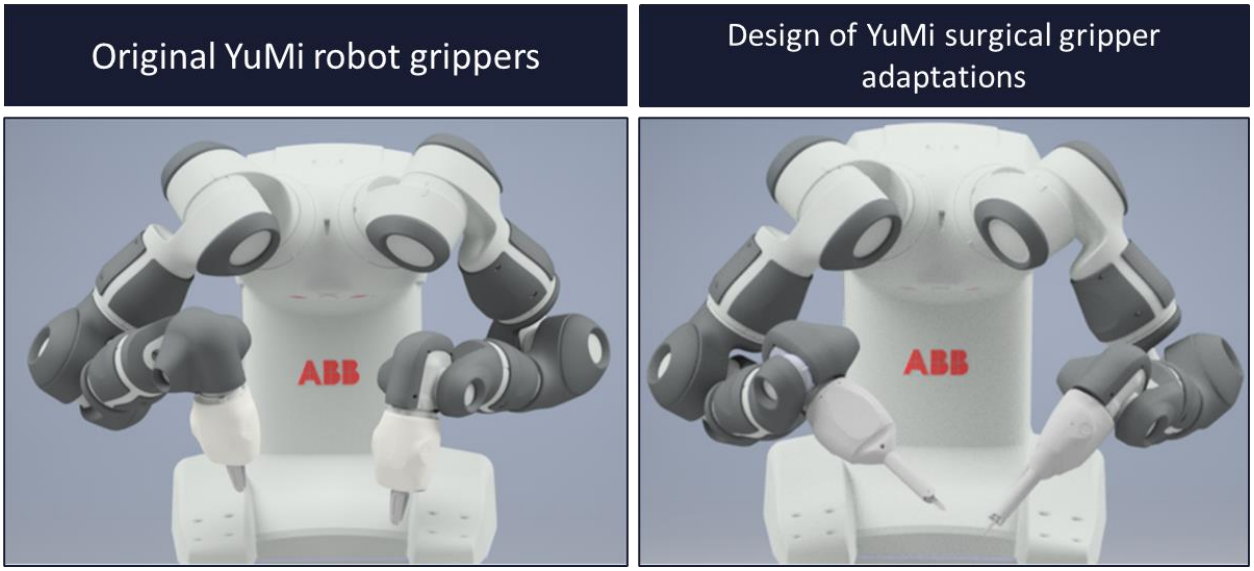

Figure 3. Original ABB YuMi Robot and gripper adaptations on the ABB YuMi robot.

The gripper was designed to adapt the YuMi robot to surgical research-related tasks. The translational degree of freedom in the original robot's parallel grip is transformed into a rotational pinch at the end-effector to operate forceps, scalpels, and scissors, among other surgical tools. The design of the gripper extension, presented in Figure 3, preserves the exterior of the original ABB robot to ensure proper enclosing of the gripper's components and safe operation. The main components of the structure are designed to be prototyped in PLA material using a low-cost fused deposition modeling 3D printer, with production costs below $20 per gripper. The 3D printer used was the open-source Prusa i3 MK2 (Prusa, 2016). The internal components of the gripper adaptations are shown in Figure 4.

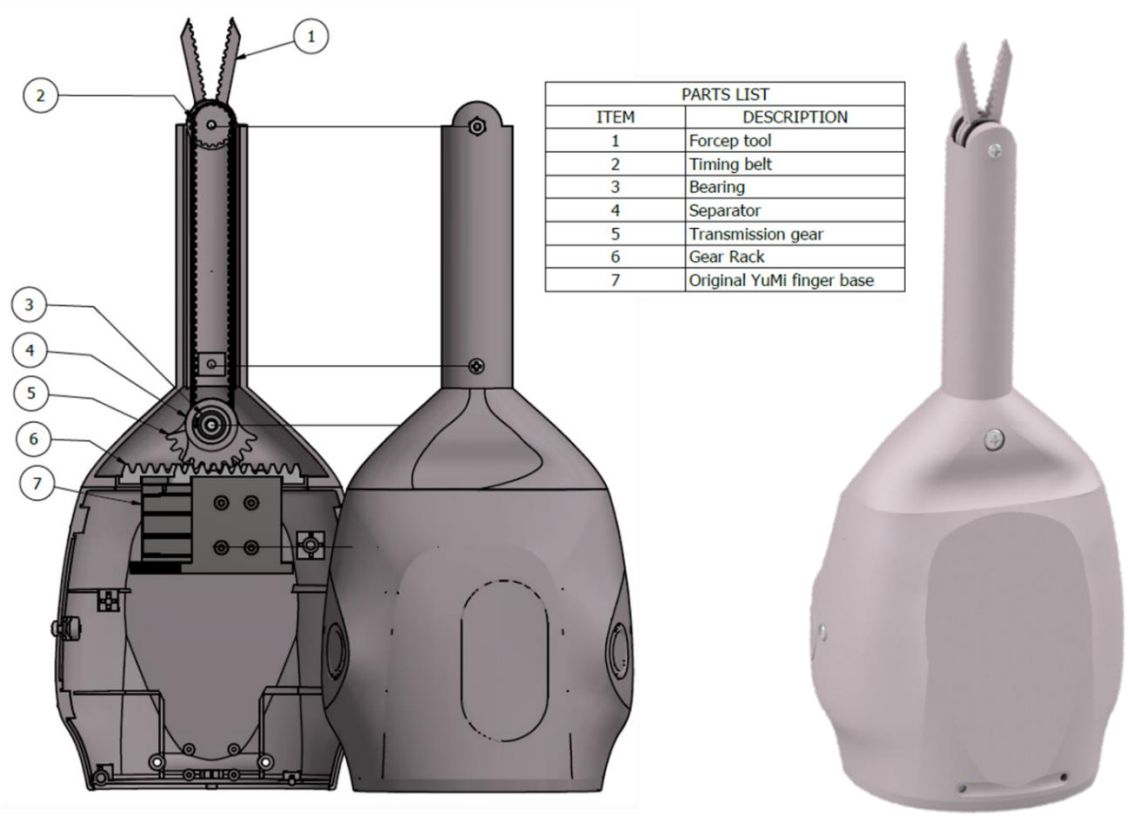

Figure 4. Cad Model of the gripper adaptation for the YuMi Robot.

A rack and pinion mechanism coupled with a timing belt were used to transmit the parallel motion of the original YuMi's finger base into a rotatory motion of the surgical tooltips. Two timing belts with 0.080" pitch, were chosen for power transmission to provide reliable rotation with no slippage on the surface. The timing belts were kept in place with radial bearings for minimum resistance and couple directly on the end-effector tooltips. The adaptations extend the tool center point of the gripper by 70 mm as shown in Figure 5.

.

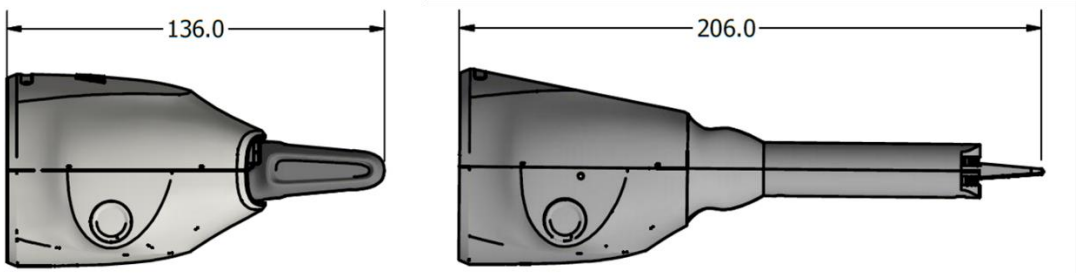

Figure 5. Length comparison of original gripper (left) and surgical gripper adaptations (right).

3.1.2 Teleoperation of the real robotic system

To achieve the adaptation of the industrial robotic platform to research in surgical robotics, we have created a system for operating the real robot using virtual reality controllers. An overview of the system architecture is shown in Figure 6.

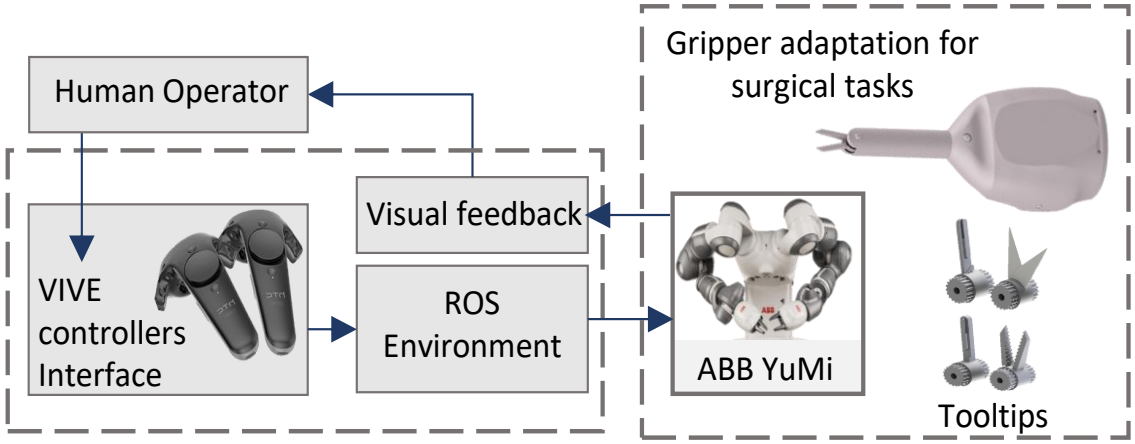

Figure 6. Overview of the robotic system

The HTC VIVE set includes a Head-Mounted Display (HMD) and two controls that provide position and orientation tracking with submillimeter precision (Niehorster et al., 2017). To control the robot, the operator used two HTC VIVE controls as the main interface (one for each hand, corresponding to each robotic arm), and can be seated in a remote location or directly beside the robot. Pose tracking of both controllers is performed with two lighthouse technology™ stations allowing the robot's end-effectors to be teleoperated simultaneously for grasping, translational motion, and orientation. Visual feedback from the surgical task can be obtained through direct observation of the robot's performance when operating in situ or additional cameras located at the top of the surgical setup.

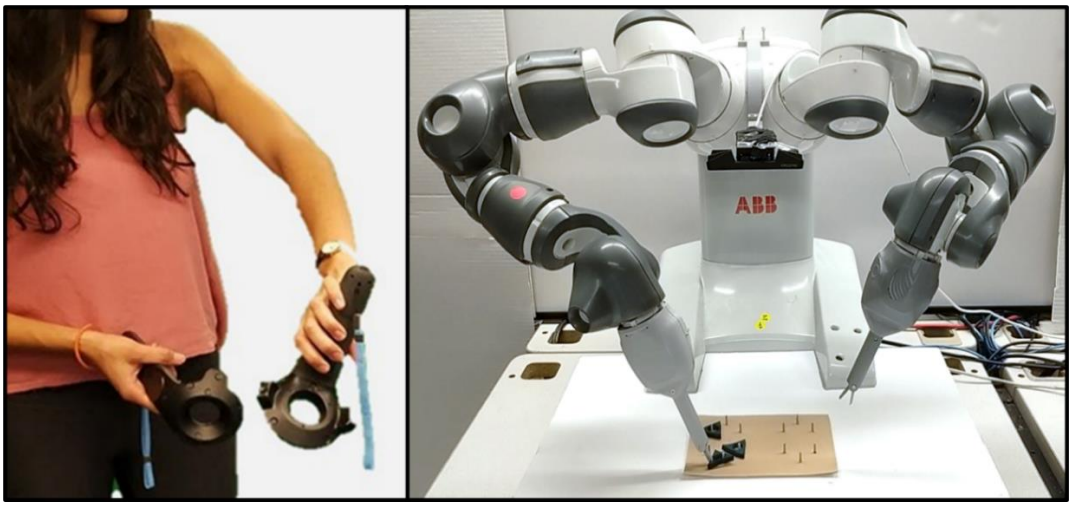

Figure 7. Teleoperation of the real robot with the HTC VIVE controllers

The HTC VIVE controls include 3 main buttons to trigger action events: the trackpad, trigger, and grip buttons shown in Figure 8. By pressing the trackpad button, the user controls the opening and closing of the gripper tooltips. When the trigger button is pressed, for each independent control, the operator drives the position and orientation of the robots' end-effectors. When the trigger is first pressed the control's reference frame is mapped to the end-effector of the robot, allowing to reset the working space by releasing the trigger.

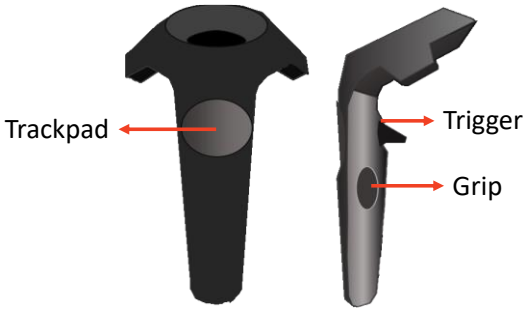

Figure 8. VIVE controls

The VIVE controllers pose and button states are sent through ROS bridge. RAPID (Robotics, 2014) is the high-level programming language used to control ABB's industrial robots. We created a package named YumiVive in the Robot Operating System (ROS) that sends RAPID commands to the YuMi Robot to control end-effector's pose and gripper state using the YuMi-python interface (Liang et al., 2017) and the robot inverse kinematics.

Gripper tools for surgical tasks

Different tools have been designed for the surgical gripper extensions, including forceps and different cutting tools, such as scalpels and scissors. For the scalpel case, two designs were proposed: in the first design the scalpel is oriented in parallel with the gripping DOF of the robot and, in the second the scalpel is oriented perpendicularly to the timing belts' actions. For both scalpel tools, 11 blades were used, which are one of the most commonly used in dermatology practice (Tobergte, 2010). In the case of scissors and forceps, the gripper's DOF is used to close the robot's tooltips allowing to cut and grasp tissue respectively. The forceps have a serrated surface design to correctly perform tissue grasping and skin retraction.

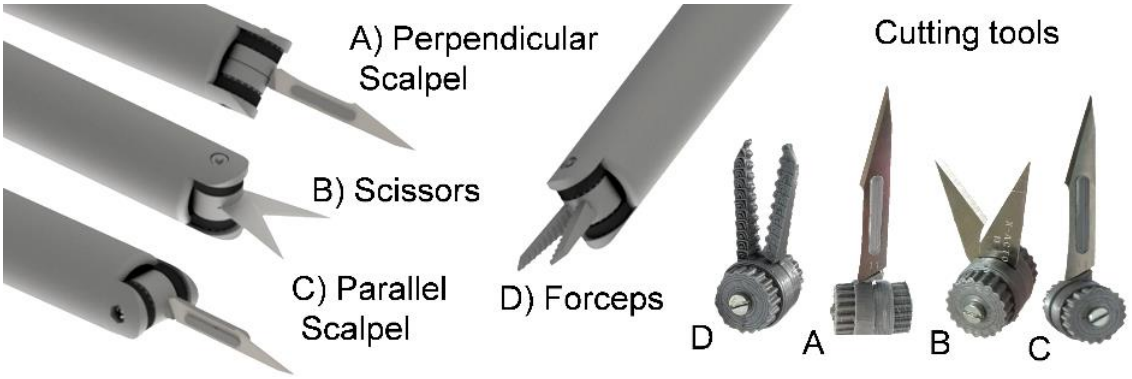

Figure 9. Different surgical designs for surgical task

Experimental evaluation of 3D printed adaptations on the real robotic system

Different tools with surgical gripper adaptations were tested in two mock-surgical procedures related to diseased tissue removal. a) melanoma extraction Figure 10.A) and, b) debridement of unviable tissue (Figure 10.B). Tissue samples of cross-linked polymer were generated from polyvinyl-acetate and sodium tetraborate. A similar polymer has been used previously for robotic surgical research (Murali et al., 2015). Preliminary results included five repetitions performed for the melanoma excision task, while three were conducted for the debridement using each cutting tool. (Sanchez-Tamayo & Wachs, 2018) Each task consists of 4 steps: approach, pinch, retract, and cut (See Figure 10).

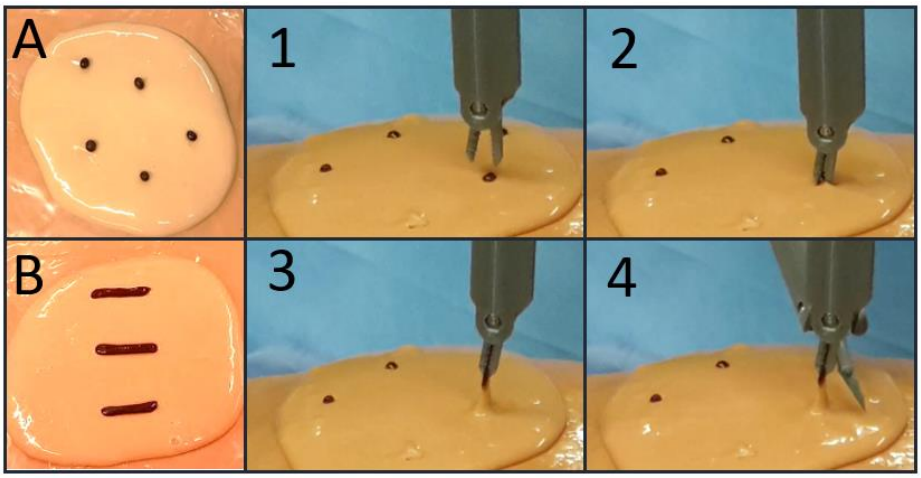

Figure 10. Steps of diseased tissue removal tasks: 1-Approach, 2-Pinch, 3-Retract, 4-Cut Action.

The effectiveness of cutting action was measured based on the percentage of healthy tissue unnecessarily removed from cut samples. Figure 11 shows the averages of healthy tissue removal plotted as bars and the standard deviations are included for each condition. Performance time was measured as the percentage of completion time for cut action for each tool, while time variable is normalized based on the longest completion time for each surgical task. The results are presented in Figure 11.

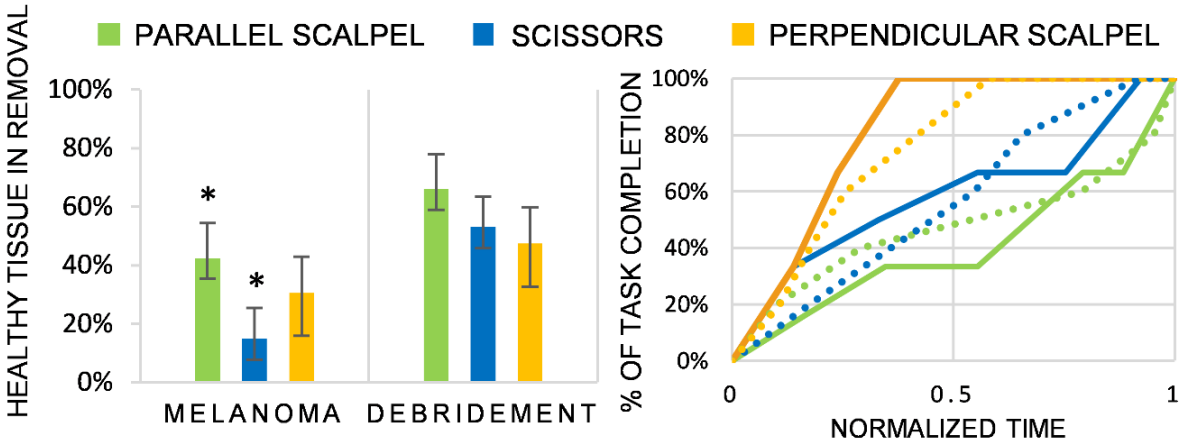

Figure 11. Left: Healthy tissue in removal. Right: Percentage of completion vs normalized time Dotted line (melanoma) solid line (debridement).

A one-way ANOVA followed by a post hoc Tukey-Kramer procedure were conducted. The result shows that the percentage of healthy tissue unnecessarily removed using scissors (15%) was significantly lower than using the parallel scalpel (42%) as cutting tool in the melanoma task

(P=0.019). The scissors tool showed the highest effectiveness in the melanoma task, while the perpendicular scalpel did so in the debridement.

In terms of performance time, the perpendicular scalpel showed the fastest completion time for both tasks while scissors and parallel scalpel showed comparable results.

The surgical gripper adaptations were tested in the real robot to evaluate the feasibility of the YuMi platform to be used for surgical robotics research. In this work, our approach was implemented and validated in a robot simulation environment. Due to time constraints, the approach was not implemented in the physical robot.

3.1.3 Robot simulation environment

We have created a robot simulation environment for the ABB YuMi robot using the CoppeliaSim robotics simulator. The designed system allows to collect trajectory demonstrations from robot teleoperation of the simulated YuMi robot using a Virtual Reality (VR) interface as well as online testing of the robot's control policies. The simulator environment was used to gather the datasets for the two tasks: the surgical debridement task and the peg transfer task. An overview of the robot simulation environment used for data collection is shown in Figure 12.

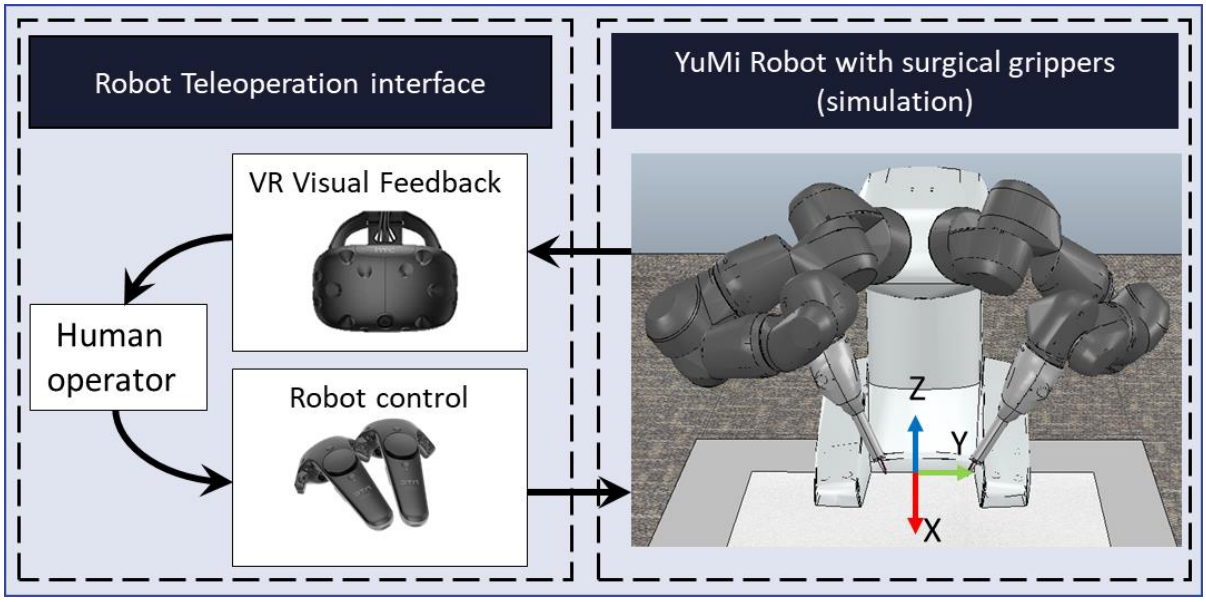

Figure 12. Overview system for data collection of task execution in simulation using virtual reality teleoperation.

The robot model

The simulated robot model was created based on the YuMi robot specifications and CAD model available in ABB's official webpage (ABB, 2020). To successfully utilize the robot for mock-up surgical tasks, the CAD model of the surgical gripper adaptations previously described has been added as the robot's end-effectors in the simulated environment. During data collection, the robot's end-effectors poses (position $\vec{p}$ and orientations $\vec{q}$) were controlled by solving the robot inverse kinematics to follow the desired poses according to the VR motion controls.

3.2 Dataset acquisition

3.2.1 Specific hardware and software

The teleoperation interface used to collect task demonstrations is based on the HTC VIVE virtual reality set. The software employed for robot simulation and data collection is the CoppeliaSim software (CoppeliaRobotics, 2020b) which offers multiple useful features such as multiple physics engines, user interaction with the environment during simulation and mesh manipulation and optimization (Pitonakova et al., 2018). The HTC VIVE can be used to interface with the simulation environment using the CoppeliaSim-VR-Toolbox by (Bogaerts et al., 2019).

3.2.2 User interface

We designed a virtual reality control interface for operating the YuMi robot intuitively in the CoppeliaSim simulation environment. The controller trackpad buttons operate the grippers individually for each robot arm and the trigger buttons enable/disables the motion of the robot. The immersive VR interface allows to map both the robot's controllers and the robot arm's pose to a common space in the virtual reality environment. When the trigger button on a robot controller is pressed, the corresponding robot arm updates its position and orientation according to the controller's motions. When the trigger is disabled, the user can move his hands freely, rest or, adjust her hand's pose for a more comfortable position without controlling the robot.

To collect multiple task demonstrations over a single session (e.g. 50), the grip button on the right controller allows resetting the state of the environment. This action places the object of interest (also called target object) in a random start position in the space (denoted with variable X_s) and

marks the start of a new trial k in the recorded dataset D_{task}. The simulation environment is displayed in the VR headset granting the user control of the robot with a realistic 3D view of the robot workspace in VR. The environment is rendered in the virtual reality headset using the CoppeliaSim-VR-Toolbox.

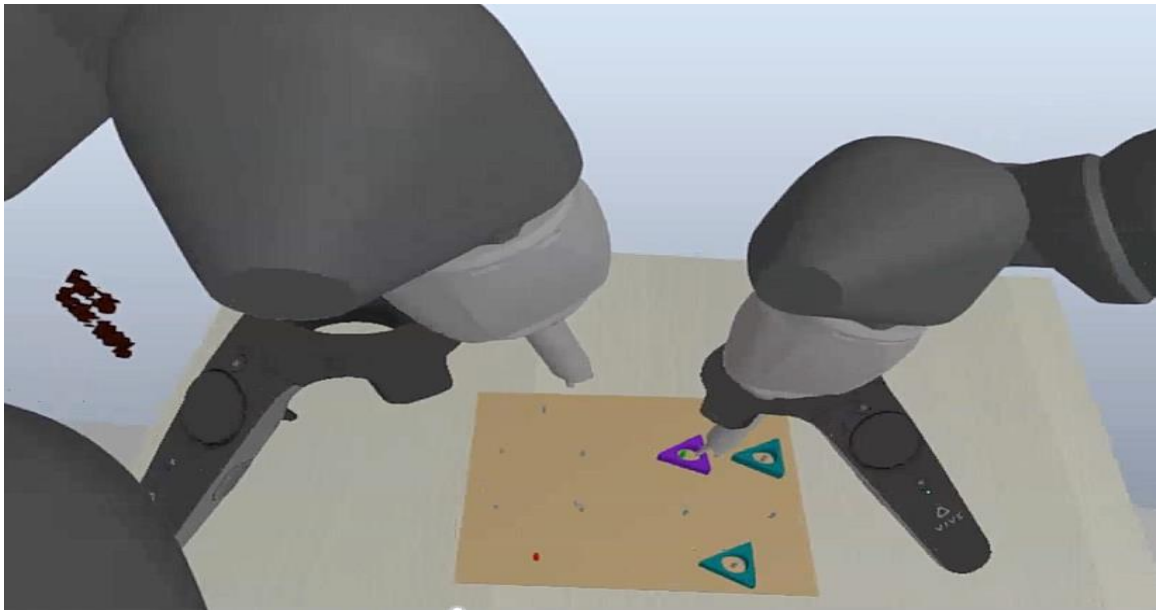

Figure 13. User view rendered in the VR teleoperation interface for the peg transfer task. The view shows the pegboard and the part to be transferred (purple).

3.2.3 Description of the tasks

Peg transfer surgical training task

The peg transfer is a task for surgical skills training included in the Fundamentals of Laparoscopic Surgery (FLS) (Ritter & Scott, 2007) that consists of the transfer of triangle shaped parts between vertical pegs over a rectangle platform. In this work, we define one episode (trial) as the transfer of one part. Transferring a part consists in picking it up from a peg with one robot arm, handing over the part to the other robot arm, moving the part over another peg, and placing it on the goal peg (Hwang et al., 2020). This procedure can be divided into a series of primitive actions or "surgemes" that compose the full task. The surgemes that compose the peg transfer have been identified as *"approach"*, *"align and grasp"*, *"lift"*, *"transfer-get arms together"*, *"transfer-exchange"*, "approach target peg" and *"align and place"* (Madapana et al., 2019) (shown in Figure 14).

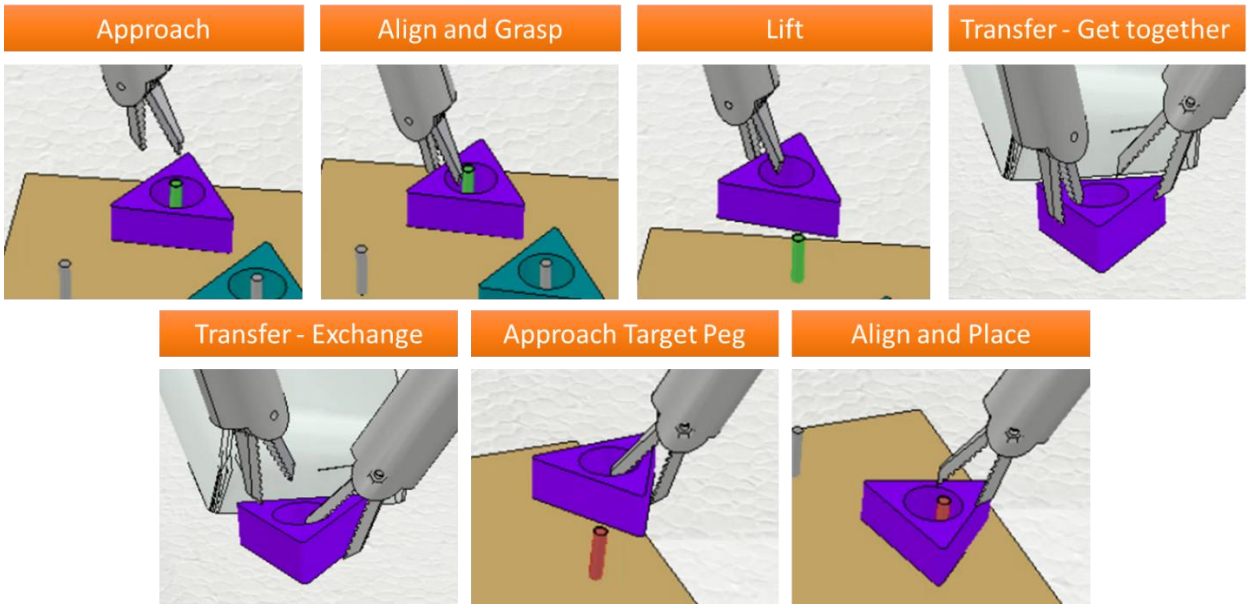

Figure 14. Surgemes for the peg transfer task.

To execute the task in simulation using the YuMi robot, a simulation environment for the peg transfer task has been created in the software CoppeliaSim. For each episode of the task, the part of interest is located on a random peg on one side of the pegboard (start) which should be transferred to a random peg on the other side of the pegboard (goal). To facilitate data collection during robot teleoperation, the start peg is shown in green to the user while the goal peg is shown in red. The environment for the simulation of the peg transfer task is shown in Figure 15.

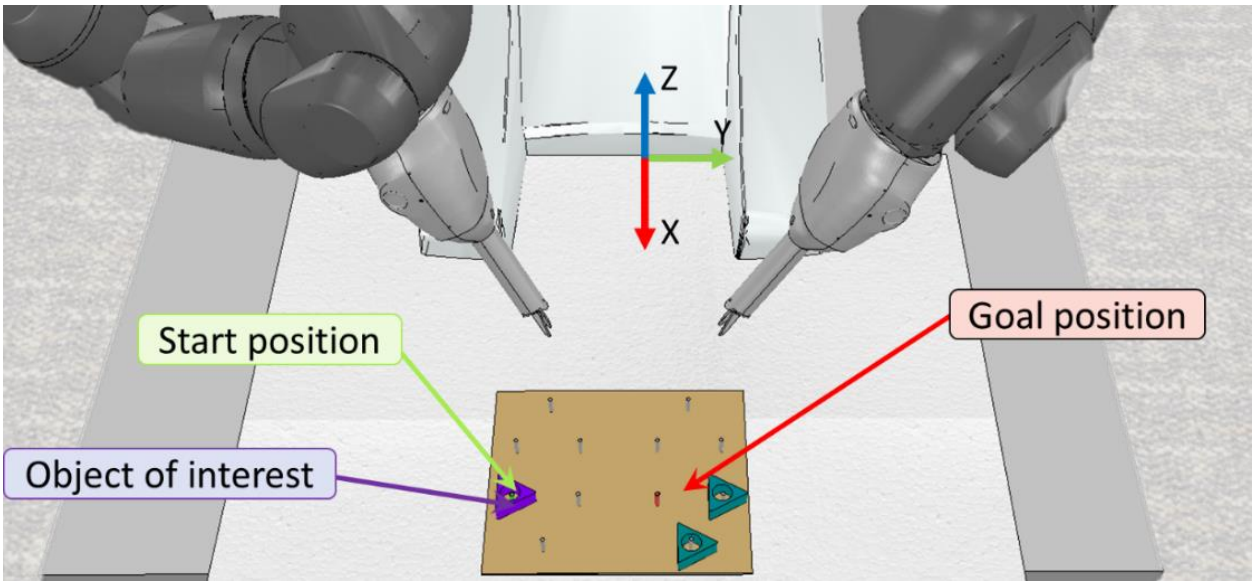

Figure 15. CoppeliaSim simulation environment for the peg transfer task.

Surgical debridement task

We have created a simulation environment that allows to simulate the debridement task, including the complexity of separating the tissue from the rest of the wound and retrieving the sample to a receptacle. The simulation environment for the debridement task is shown in Figure 16, the goal position of the debrided fragment is the location of the tray, denoted as X_g.

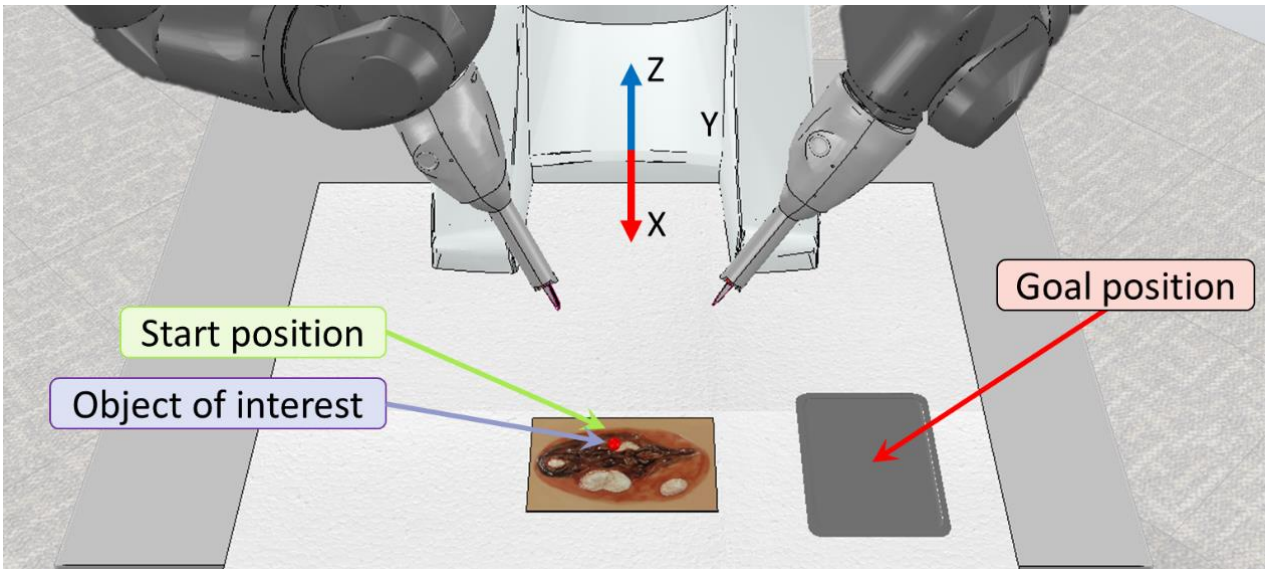

Figure 16. CoppeliaSim simulation environment for the surgical debridement task.

In our work, the debridement task involves picking up the diseased fragment from the wound, retracting the tissue to perform a cut action to separate the tissue fragment from the healthy tissue, and a final step where we deposit the tissue sample on a surgical tray. Based on this description, the key surgemes of the task can be identified as *"approach", "grasp*", "*retract*", "*cut*", and "*place on tray*" (shown in Figure 17). For each episode of this task, the tissue fragment is placed on a random location over the simulated tissue, and the tray is located on either side of the robot at a random distance from the robot base (in the x-axis).

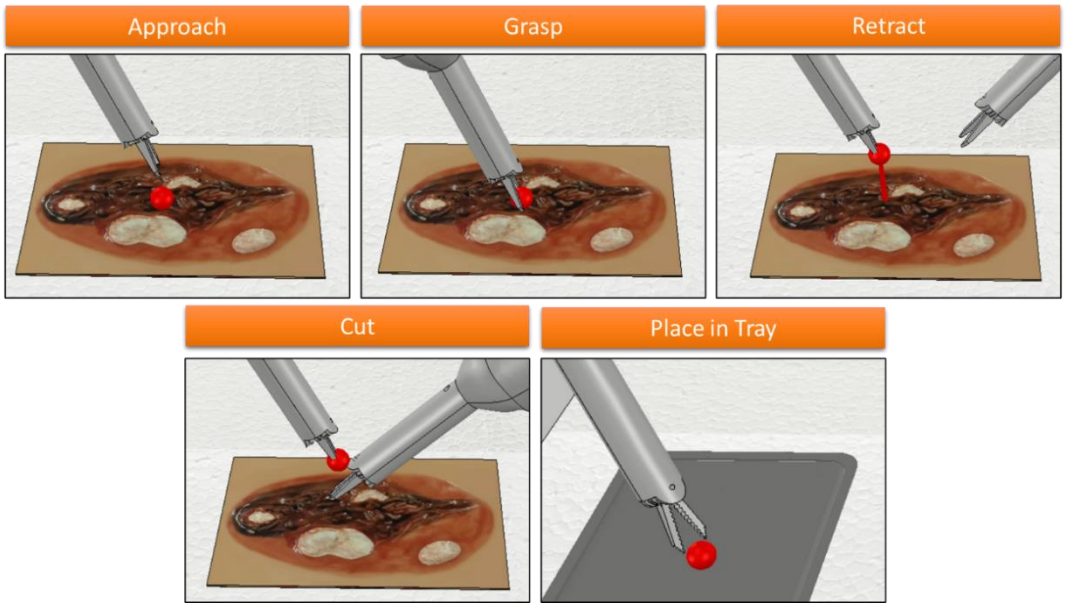

Figure 17. Surgemes that compose the surgical debridement class.

3.2.4 Data description

During data collection, the simulation environment includes trajectory demonstrations of the task recorded at a rate of 20Hz (recommended simulation rate in the CoppeliaSim simulation environment). Each trajectory demonstration τ is constituted by a series of features $\emptyset$ (one for each point of the trajectory), related to the robot pose $\{\vec{p}, \vec{q}\}$ and gripper state g, as well as the state of the environment e. The features recorded in the simulation environment are described on Table 1. The poses (position and orientation) are recorded with respect to coordinate space of the robot base U. In this work, all orientations are represented using unit quaternions.

Table 1. Features included in the dataset. The subscripts ***r*** and ***l*** represent the right and left arms of the robotic system.

Features recorded		
Trial #	k	Trial id
Timestamp	t	Seconds
Left and right end-effector orientation, respectively	$\vec{q}_l(t), \vec{q}_r(t)$	$[r_x, r_y, r_z, r_w]$
Left and right end-effector position, respectively	$\vec{p}_l(t), \vec{p}_r(t)$	$[p_x, p_y, p_z]$
Right and left gripper state	$g_r(t), g_l(t)$	open/close
Left and right robot joints state	φ_l, φ_r	$(\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6, \varphi_7)$
Object of interest position	$\vec{o}_p$	$[p_x, p_y, p_z]$
Start position	X_s	$[p_x, p_y, p_z]$
Goal position	X_g	$[p_x, p_y, p_z]$
Pegboard position	b_p	$[p_x, p_y, p_z]$
Pegboard orientation	b_q	$[r_x, r_y, r_z, r_w]$
Dominant side	d_s	right/left

The method presented in this work can be generalized to different mock-up surgical tasks by identifying the key variables in the task, which are the object of interest o_p (target object), and the "start" X_s and "goal" X_g position. The environment state e is captured by recording the positions of the objects of interest according to the task at hand. In the peg transfer task, the object of interest is the part to be transferred between pegs. In the peg transfer task, the "start position" is the position of the peg where the part starts and the "goal position" is the position of the peg where the part needs to be placed. In the surgical debridement task, the object of interest is the fragment of tissue that needs to be extracted. For this task, the "start position" is the starting position of the fragment, and the "goal position" is the position of the tray where the fragment needs to be deposited. The "dominant side" feature is defined as the side of the robot that is dominant for the task. For example, in the peg transfer task the dominant side is "left" when the peg transfer is performed from left to right. For the debridement task, the dominant side corresponds to the side of the robot where the surgical tray is located.

3.2.5 Data collection

In the simulation environment, the user can start the data collection by pressing the right-hand gripper button, which resets the environment to a random state and marks the start of a new trial *k*. During teleoperation, the operator can mark a trial as a failure by pressing the grip button on the left arm and continue the data collection with a new trial. This system allows to continuously collect numerous robot teleoperated demonstrations as the system can automatically reset the environment and automatically annotate the data. In addition, the system allows to flag failed attempts to avoid the LSTM policy to learn undesired behavior. A trial marked as "fail" are those that include any of the following: failure to hold the part (dropping the part unintentionally), collisions between the robot arms, collisions with the surface, or failure to properly deposit the part at the goal location.

During data collection, variability was added to the data by changing the initial state of the environment to a random initial configuration. For the peg transfer task, the board was placed at a random position following a uniform distribution with 10 cm of variation: $b_{py} \sim U(-5,5)$ cm $b_{px} \sim U(-5,5)$ cm from the center of the robot workspace. This distribution was selected to provide sufficient variation of the setup while keeping the center of the board in an area where both grippers can reach with ease.

The board was randomly rotated around the z-axis with a rotation angle given by a uniform distribution of $\sim U(-15°, 15°)$. Further variations of the peg transfer task were obtained by randomly choosing any peg from each side of the pegboard, as the start and goal peg, respectively. The side from which the peg transfer task starts (left or right) was chosen at random as well. In the debridement task, variation was introduced by randomizing the position of the tissue part to be removed around the whole surface of the debridement simulator. To cover the surface of the debridement tissue simulator (15x10 cm), a uniform variation of $o_{py} \sim U(-7.5,7.5)$ cm, $o_{px} \sim U(-5,5)$ cm was chosen. The surgical Mayo tray can either be located on the left side of the robot or the right side, at a random distance in the x axis: $\mathcal{G}_{px} \sim U(-10,10)$ cm. This variation allows the Mayo tray to cover the work surface of the robot with a clearance of 5 cm from the edges of the surface or the base of the robot.

A human operator collects over 1000 observations for each of the two tasks using the aforementioned data collection procedures. Once collected, the demonstrations were filtered to eliminate the failed trials. A summary of the collected datasets is presented in Table 2, only the successful demonstrations were included in the dataset resulting in 839 successful demonstrations obtained from peg transfer and 934 from debridement.

Table 2. Summary of the collected datasets.

Task	Peg transfer	Debridement
Number of full task demonstrations	839	934
Number of points in demonstrations	269,466	532,201
Average demonstration length (waypoints)	321.18	569.81

The original teleoperated demonstration recorded at 20Hz were subsampled at a lower frequency to train the LSTM for robot manipulation more efficiently as suggested in literature (R. Rahmatizadeh et al., 2018) More trajectories can be created from higher frequency recorded demonstrations as shown in Figure 18. The subsampling rate was varied as a hyperparameter during training and the best results were obtained using a 5Hz subsampling rate.

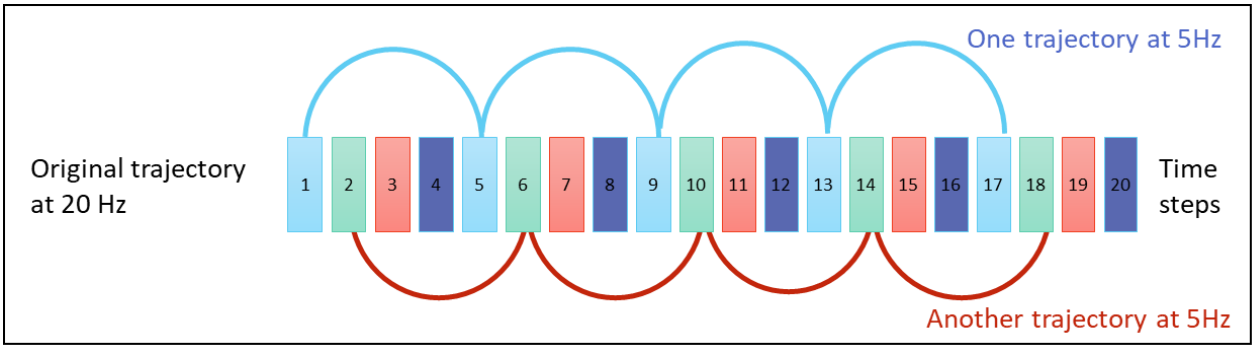

Figure 18. Demonstrations of multiple trajectories sampled at 5Hz from the original trajectories obtained at 20Hz.

3.3 Formulation of the imitation learning problem

In this section, the imitation learning problem is formulated. The objective of imitation learning is to learn a control policy for task execution that replicates the behavior of demonstrations provided by a human teacher. The human operator records a series of trajectories $\tau = [\emptyset_0, \ldots., \emptyset_T]$ formed

by features $\emptyset$, where T is the total number of time steps that constitute the demonstration of one task episode (trial). The features recorded encompass the state of the robotic system s_t and the state of the environment e_t at every time step t. The state of the robotic system is given by the pose of the robot's end-effectors and the grippers' state $P_l = \{p_l, q_l, g_l\}_t, \mathrm{P}_r = \{p_r, q_r, g_r\}_t$ for the left and right gripper, respectively. The state of the environment at time step t is given by the position of the object of interest $\{o_p\}_t$, the start position $\{X_s\}_t$, the goal position $\{X_g\}_t$ and the dominant side $\{d_s\}_t$.

$$s_t = \{q_l, q_r, p_l, p_r, g_l, g_r\}_t, \quad e_t = \{o_p, X_s, X_g, d_s\}_t \tag{5}$$

Based on the demonstrated trajectories τ, a dataset of demonstrations $D_{task} = \{(o_t^{(i)}, a_t^{(i)})\}$ is formed as a set of observations o_t paired with a corresponding action a_t. The observation o_t at each time step t is defined as the sequence of state of the robotic system s_t and state of the environment e_t for the previous n time steps. The action a_t is defined as the state of the robotic system at the next time step.

$$o_t = \{\, s_{(t-n:t)}\, , e_{(t-n:t)}\}, \quad a_t = s_{t+1} \tag{6}$$

The LSTM control policy π_θ is trained to predict the action a_t based on the current observation o_t. The policy is trained as a supervised learning problem by minimizing the mean square error loss function $\ell(a_t)$ of the network action prediction for the next time step a_t based on the dataset D_{task}. The input to the network is defined as the follow observation:

$$o_t = \{\, s_{(t-n:t)}\, , e_{(t-n:t)}\} = \{q_l, q_r, p_l, p_r, g_l, g_r, o_p, X_s, X_g, d_s\}_{t-n:t} \tag{7}$$

Each observation at time t includes the pose of the end-effectors' tooltips and the gripper states from the previous n time steps, as well as the position of the target object, the start position, the goal position, and the dominant side.

During execution, the policy π_θ generates the robot control action $a_t = \pi_\theta(o_t)$ given the current observation o_t. Examples of current control actions are the position, orientation, and gripper state (open/close) of both arms. The control action a_t, is used to control the pose of each end-effector and the grippers' state at the next time step $\{p_l, q_l, g_l\}_{t+1}, \{p_r, q_r, g_r\}_{t+1}$.

$$a_t = \{q_l, q_r, p_l, p_r, g_l, g_r\}_{t+1} \tag{8}$$

3.4 LSTM control policy for dual-arm tasks

3.4.1 Network architecture

The architecture of the network used is shown in Figure 19. The network is formed by a three-layer staked Long-Short Term Memory (LSTM) with skip-connections that provides an additional connection of the input with the second and third LSTM layers. The neural network control policy is learned for the whole multistep task as opposed to modeling each subtask or "surgeme" independently. The inputs to the network, o_t include the state of the robotic system and the state of the environment for the *n* most recent steps. The output of the LSTM network is the robot arms' pose and gripper state for the next robot command action.

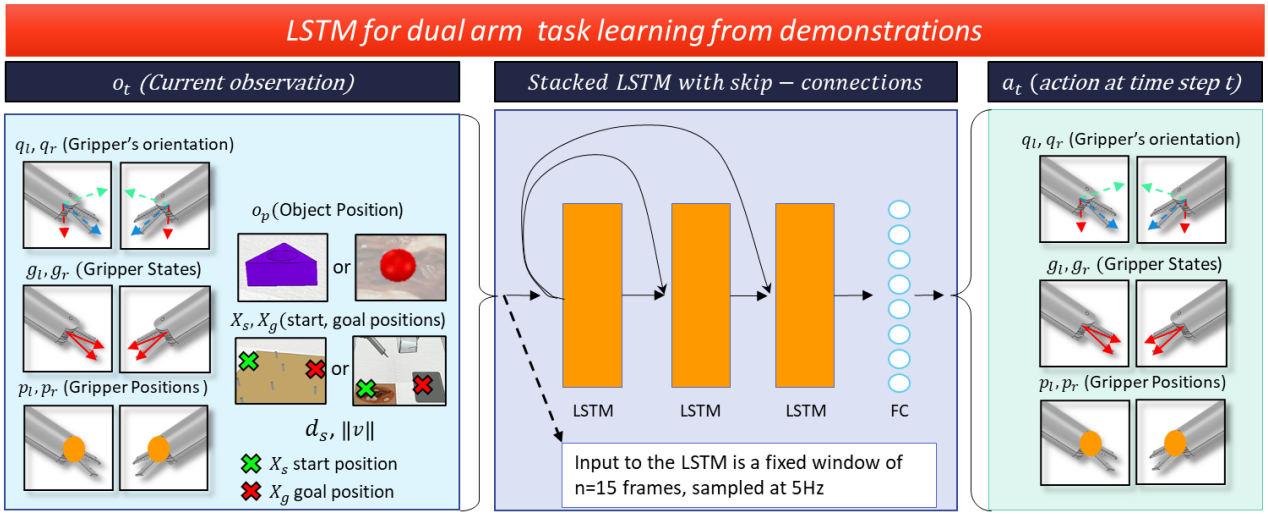

Figure 19. LSTM skip-connection architecture for robot task learning. Three-layer stacked LSTM network takes as an input the n=15 most recent states of the system to predict the next robot action for each arm.

The network learns a policy that controls both arms jointly by concatenating the features of each robot arm in the inputs o_t and outputs a_t of the network.

$$o_t = \{q_l, q_r, p_l, p_r, g_l, g_r, o_p, X_s, X_g, d_s\}_{t-n:t}\ , a_t = \{q_l, q_r, p_l, p_r, g_l, g_r\}_{t+1} \tag{9}$$

The state of the robotic system is represented by the pose of each robot arm (position and robot gripper orientation in quaternion) as well as the gripper state (open/close). The state of the

environment is represented by the position of the object of interest according to the task. The object of interest is the triangle shaped part in the peg transfer task and a tissue fragment to be debrided in the debridement task. In addition to the variable features in the environment, the network takes as an input the start positions of the object of interest, its goal position, and the dominant side feature.

3.5 Facilitating learning through affine transformations to a unified coordinate space

Transforming highly variable demonstrations to a coordinate space where key points of the task are aligned can facilitate learning by reducing the exploration space in the coordinate system where the policy is learned. In this section, we present a formulation that relies on the use of affine transformations to map the trajectory demonstrations to a coordinate space representation where the network policy can be trained with higher sample efficiency. Moreover, the proposed approach maintains the applicability of the control policy to settings with high variability.

For each episode of the task, an affine mapping can be defined to project the observations from the original robot space to a unified robot coordinate space where the start and goal positions of the object of interest are aligned to the same positions in the 3D space. Such mapping can transform large variations in the original robot space into a structured representation of the dataset that facilitates learning of the neural network.

For each demonstration in the dataset, a unique affine transformation Ψ is defined to map the observations o_t and action pairs a_t from the original robot coordinate space U to the unified robot coordinate space U'. As previously shown in equations (2) and (4). The dataset represented in the unified robot space D'_{task} is then used to train the neural network policy. During test time, an analogous affine transformation is applied to the current observation to evaluate the network policy and obtain the robot control command in the unified space a'_t. Then, the inverse mapping is used to transform the output of the network to the original robot space a_t, which is sent to the robotic system as the next control command. The use of affine transformations within the architecture of the LSTM control policy is shown in Figure 20.

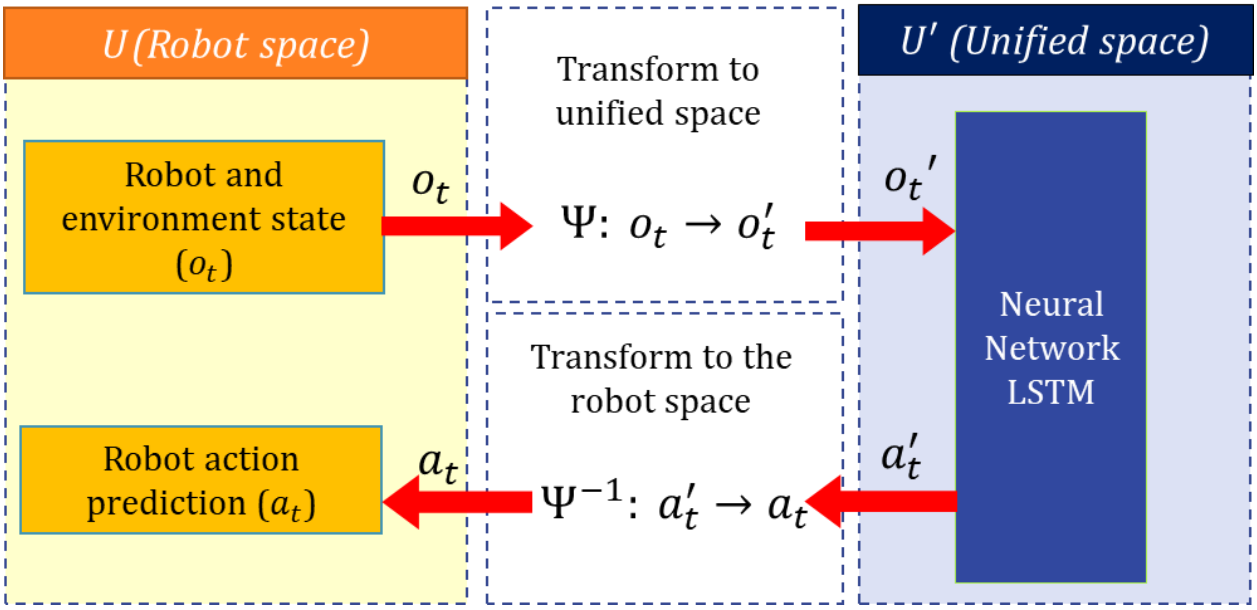

Figure 20. Transformation of the environment state to the reduced space for action prediction using LSTM and inverse transformation of the action to the original space

Affine transformations are transformations that preserve collinearity and are generally composed of rotations, translations, scale and shear (Weisstein, 2020). The affine transformation homogeneous matrix is of the general form:

$$M = \begin{bmatrix} m_{00} & m_{10} & m_{20} & t_x \\ m_{01} & m_{11} & m_{21} & t_y \\ m_{02} & m_{12} & m_{22} & t_z \\ p_i & p_i & p_i & w \end{bmatrix} \tag{10}$$

Where t_i represents translation, p_i are projections, w represents the homogeneous component and $m_{i,j}$ represent rotation, scale, and shear.

In our work, we use an affine transformation composed by three components: scale, rotation, and translation. The scaling component of the affine transformation scales the X-Y plane to map the distance between the start and goal in the robot space to the distance of the start and goal expressed in the unified space. The Z-axis is not scaled. The scale transformation is followed by rotation and translation transformations that align the goal and start points from the demonstration to the goal and start points in the unified robot space.

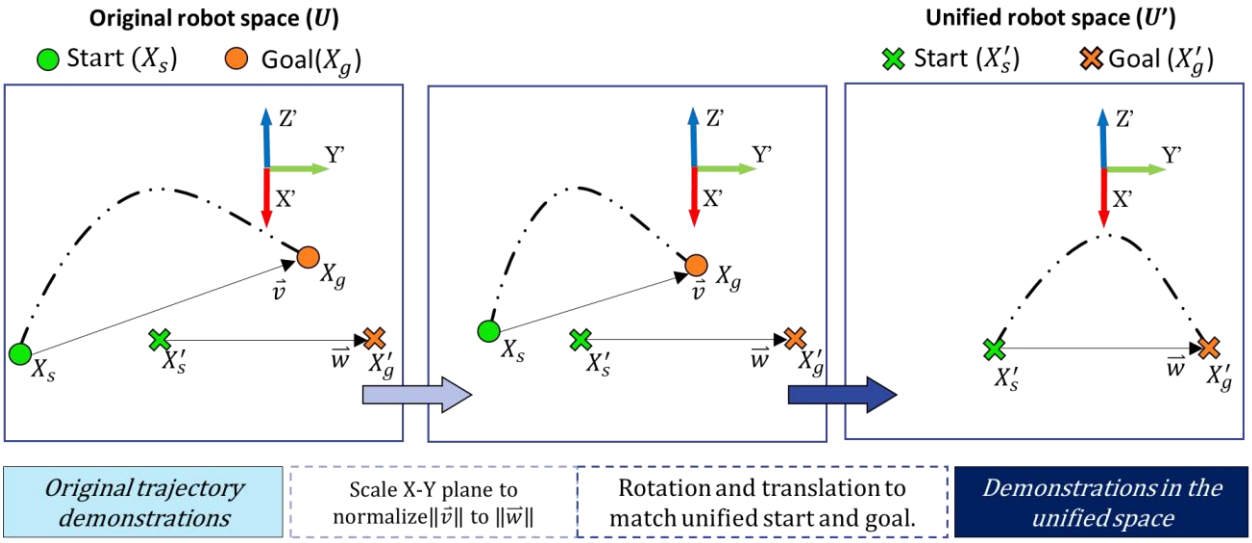

Figure 21. Affine transformation (scale, rotation, and translation) to transform robot trajectories to a unified robot space where source and target of all demonstrations are aligned to the same points in the 3D space.

Let $\vec{v}$ be defined as the vector from the start to the goal in the original robot space, and let $\vec{w}$ be the vector from the start to the goal in the unified robot coordinate space (as shown in Figure 21). The affine transformation can be defined as the set of transformations to move $\vec{v}$ into $\vec{w}$, which maps the original robot coordinate space U into the unified robot coordinate space as U'.

First, the centroids in both coordinate spaces $Q_\chi, Q_{\chi'}$ are found as the average of the start and goal positions.

$$Q_\chi = \frac{1}{2}(X_s + X_g) \tag{11}$$

$$Q_{\chi'} = \frac{1}{2}(X'_s + X'_g) \tag{12}$$

The scale transformation scales the X-Y plane so that $\|\vec{w}\| = \|\vec{v}\|$. This is obtained by scaling the plane by a factor S_{factor} in the direction of $\hat{v}$. S_{factor} is given by:

$$S_{factor} = \frac{\|\vec{w}\|}{\|\vec{v}\|}, \hat{w} = \frac{\vec{w}}{|\vec{w}|}, \hat{v} = \frac{\vec{v}}{|\vec{v}|} \tag{13}$$

The scale matrix is defined by (Goldman, 1990) as:

$$S = \begin{bmatrix} I - (1 - S_{factor})(\hat{v}\hat{v}^T) & 0 \\ ((S_{Factor}Q_\chi) \cdot \hat{v})\hat{v} & 1 \end{bmatrix} \quad (14)$$

Where the centroid of the original space Q_χ is the scaling origin and I denotes the identify matrix.

The rotation matrix R, is defined as the matrix that rotates the unit vector $\hat{v}$ an angle θ into unit vector $\hat{w}$. Let $\hat{k}$ denote the unitary vector describing the axis of rotation about which $\hat{v}$ rotates. $\hat{k}$ is the cross product of the vectors $\hat{v}$ and $\hat{w}$ which determine the plane of rotation.

$$\hat{k} = \frac{\hat{v} \times \hat{w}}{|\hat{v} \times \hat{w}|} = \frac{\hat{v} \times \hat{w}}{\sin\theta} \quad (15)$$

$$Let\ H = \hat{v} \times \hat{w} = \hat{k}\sin\theta \quad (16)$$

The rotation matrix can be found using the Rodrigues formula (Murray et al., 1994):

$$R = I + [K]_\times \sin\theta + [K]^2{}_\times (1 - \cos\theta) \quad (17)$$

Where $[K]_\times$ is the skew-symmetric cross-product matrix of $\vec{k}$.

$$[K]_\times \stackrel{\text{def}}{=} \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & k_1 \\ -k & k_1 & 0 \end{bmatrix} \quad (18)$$

Expressing the rotation in terms of $\hat{v}$ and $\hat{w}$, the rotation matrix is given by:

$$R = I + [H]_\times + [H]^2{}_\times \frac{1 - \cos\theta}{\sin\theta^2} \quad (19)$$

Where:

$$\sin\theta \stackrel{\text{def}}{=} \frac{|\hat{v} \times \hat{w}|}{|\hat{v}||\hat{w}|}, \quad \cos\theta \stackrel{\text{def}}{=} \frac{\hat{v} \cdot \hat{w}}{|\hat{v}||\hat{w}|} \quad (20)$$

$$[H]_\times = [K]_\times \sin\theta \quad (21)$$

$$[H]^2{}_\times = [K]^2{}_\times \sin\theta^2 \quad (22)$$

For tasks where the source and target are not co-planar, the unitary vectors $\hat{v}$ and $\hat{w}$ should be defined projected over the X-Y plane.

The translation matrix is given by

$$\mathrm{T} = \begin{bmatrix} \mathrm{I} & \Delta_\mathrm{p} \\ 0 & 1 \end{bmatrix} \quad (23)$$

Where $\Delta_p \; is \; given \; by$:

$$\Delta_p = Q_{\chi'} - RQ_{\chi} \tag{24}$$

Finally, the scale, rotation, and translation matrices can be combined to produce the affine transformation Ψ given by the matrix M:

$$M = T * R * S = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix} \tag{25}$$

The matrix M in its homogeneous form transforms all position features to the unified robot coordinate space by matrix multiplication.

$$\vec{p}' = \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{bmatrix} m_{00} & m_{10} & m_{20} & t_x \\ m_{01} & m_{11} & m_{21} & t_y \\ m_{02} & m_{12} & m_{22} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{26}$$

Similarly, at prediction time the inverse matrix M^{-1} is used to transform the robot command computed at the output of the network into the original robot space for robot execution.

$$M^{-1} = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix}^{-1} \tag{27}$$

All orientation features are first transformed from quaternion to rotation matrices, where the rotation transformation is applied and then transformed back to quaternion form:

$$q_i \rightarrow R_i \tag{28}$$

$$R'_i = RR_i \rightarrow q'_i \tag{29}$$

As defined, applying the affine transformation results in a dataset where the start and goal position of the object of interest are aligned. Figure 22 shows two trajectory demonstrations for the peg transfer task before and after applying the affine transformation mapping. An additional feature $\|v\|$ is added to the network input to facilitate learning of commonalities in the original space. In the peg transfer task for example, this feature could be important for the network to learn how far from the peg it needs to approach the object to successfully grasp it.

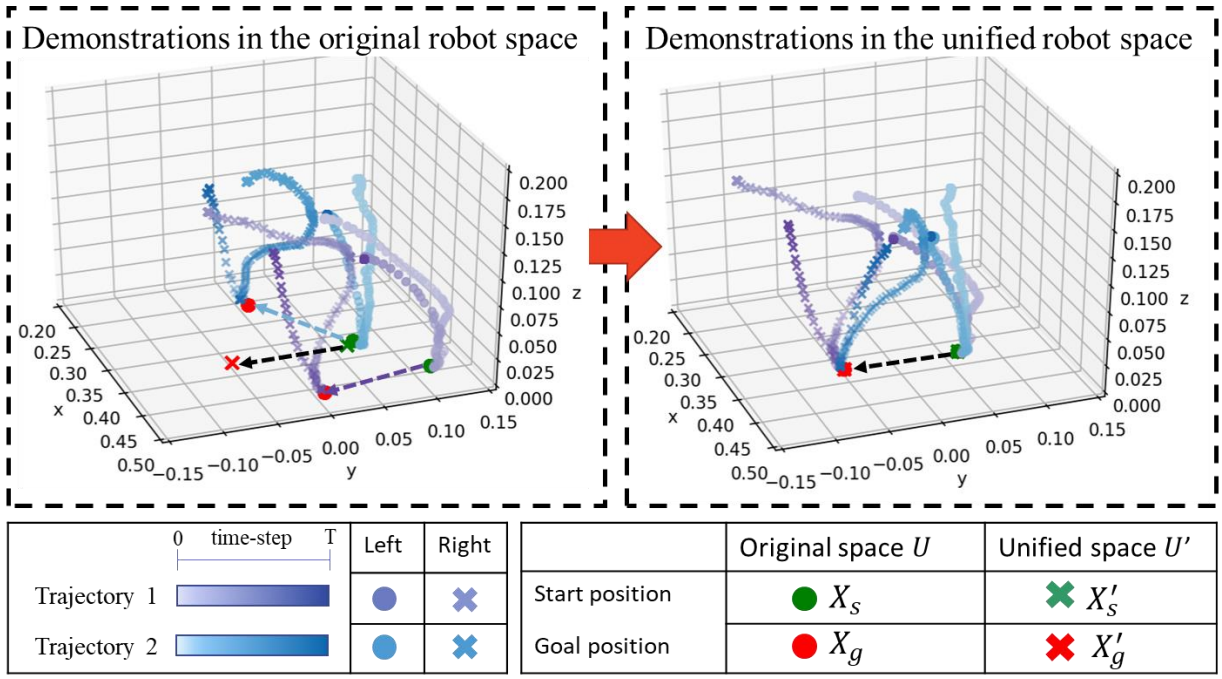

Figure 22. Example of two recorded demonstrations in the original robot space and their equivalent representation in the unified robot space. The demonstrations belong to the dataset of the peg transfer task.

3.6 Training the network control policy

The robot control policies are trained in Python using the Keras functional API (keras.io, 2020) on top of the Tensorflow framework. Keras is a high-level application programming interface for building and training deep learning models in python (Ramasubramanian & Singh, 2019). Tensorflow is an interface for the implementation of machine learning algorithms built by google (Martín Abadi et al., 2015). The robot control policies are tested on the robot simulator using the CoppeliaSim Python remote API (CoppeliaRobotics, 2020a).

We used two networks for learning the peg transfer and the debridement tasks, respectively, with similar architectures. Each network was trained applying the mean square error loss function mse on the prediction of the next robot state: a_t. The Adam optimization algorithm was used with an initial learning rate l_r of 0.001 and all the parameters of the network were initialized using the glorot uniform initializer (Glorot & Bengio, 2010). The network was trained for 1000 epochs with early stopping when the loss during validation did not improve for 200 epochs. In addition, the learning rate l_r was reduced by 0.3 if the validation loss plateaued for 100 epochs. From the

collected dataset, 80% of the demonstrations were used for training and 20% were used for validation and hyperparameter tuning.

A MinMax scaler was used to map all the inputs to the range [0,1]. The positions in the cartesian space are scaled based on the same MinMax for all axis components (x,y,z). The limiting values for the position were determined based on the maximum and minimum values of the X,Y and Z coordinates in the dataset (as shown in Table 3). All quaternions have been normalized to fit the range [-1,1].

Table 3. Min-max scaler limits for the inputs of the network.

Feature type	Min/Max scaler
$\boldsymbol{positions}$ $(\boldsymbol{p_x, p_y, p_z})$]	[-0.5,0.9]
quaternions $[\boldsymbol{r_x, r_y, r_z, r_w}]$	$[-1,1]$

In our LSTM architecture, each one of the stacked LSTM layers received as input the output of the previous layer concatenated with the input to the network using skip-connections. During training, different hyperparameters were considered such as the subsampling frequency of the data, the number of hidden nodes in each layer, dropout after each layer, initial learning rate and number of epochs. Although the subsampling frequency of the training data was varied during hyperparameter search, the rate of evaluation of the policy was maintained at 4Hz during validation and testing for consistency.

For the debridement task, the best results were obtained with 3 stacked LSTM layers composed of 256 hidden nodes each. For the peg transfer task, the best results were obtained with 3 stacked LSTM layers with 700 hidden nodes each. In order to increase generalization performance, gaussian noise was added to the input (0.01) and to the hidden layers (0.1) followed by batch normalization. The noise is a regularization layer and it is only active during training. In addition, dropout (0.25) was added after each LSTM layer to reduce overfitting.

3.7 Testing the network control policy

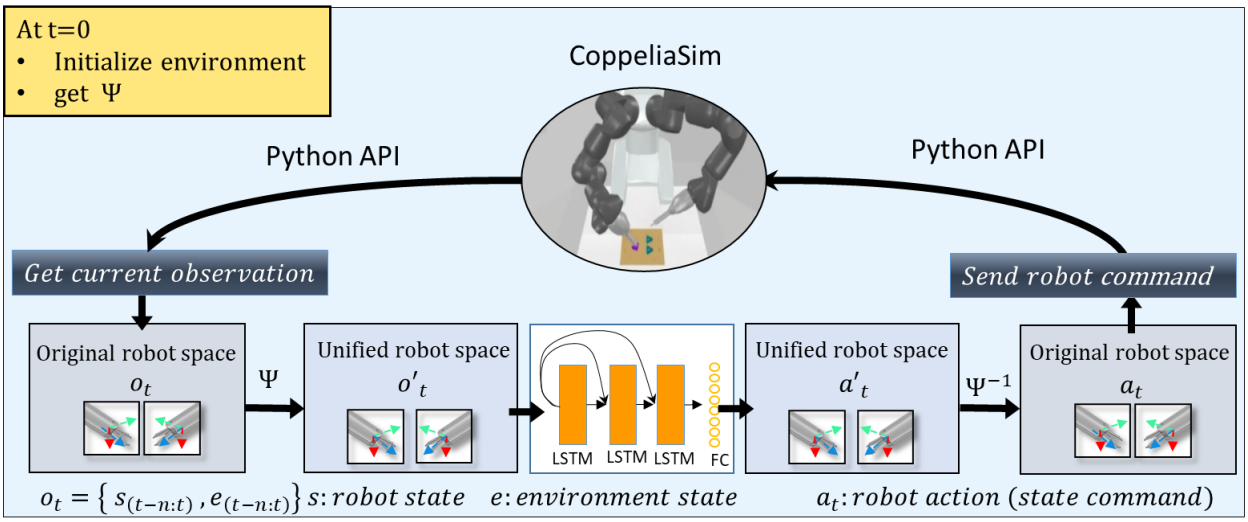

Figure 23. Evaluation of the LSTM control policy in simulation.

In order to validate the LSTM control policy in an imitation learning setting, the performance of the network was evaluated during task execution in a simulated scenario. The state of the environment was initialized in a new random configuration for the task as was done during data collection. The online evaluation of the robot control policy takes as an input the observation o_t which includes the state of the robot s_t and the objects of interest in the environment e_t for the n=15 most recent time steps. The output of the network is the prediction of the robot arm's poses and the status of the grippers $\{q_l, q_r, p_l, p_r, g_l, g_r\}_{t+1}$ at time t+1, which is sent as a robot command a_t. Based on the robot command sent, the inverse kinematics configured in the simulated environment was used to solve for the robot positions required to reach the desired end-effector poses.

The robot control policy was evaluated using the CoppeliaSim Python remote API. At each time step, the python script sent a robot command to control both robot arms and waited for a reply from the simulation environment that included the new robot and environment state $\{s_{t+1}, e_{t+1}\}$. The robot control policy was evaluated at an average sample rate of 4Hz, which is sufficient for the robot to reach the predicted pose and retrieve the new robot and environment states from the simulation environment. For each testing trial, the policy was followed, frame by frame, until the

task was completed or to a maximum of 160 time step iterations. For consistency, the state of the robot was reset to the home position after each episode of the task.

3.8 Discussion

The proposed control policy implicitly optimizes for both the arms and the whole task as opposed to modeling each arm and subtasks independently. In the context of surgical robotics research, the method can be used to learn mock-up surgical tasks formed by several steps without the need to segment the demonstrations is surgical gestures. Moreover, the proposed architecture allows the LSTM network to jointly learn the control of both arms working in collaboration by concatenating the features for both end-effectors in the network's input and output. Furthermore, the affine transformations can be employed to facilitate learning of multiple manipulation tasks by identifying two key points of the task ("start" and "goal"). For example, in a "push to pose" task the start position of the object can be defined as the "start" point and the desired position of the object as the goal. For tasks that have only one evident key position in the space such as a "grasping" task with no clear "goal", the "goal" point can be replaced for a fixed point in the robot coordinate space such as the base of the robot.

3.9 Summary

This chapter formulates the problem of imitation learning for learning dexterous dual-arm tasks in the context of Robotic-Assisted Surgery (RAS) research. An approach was proposed for learning dual-arm control policies from robot teleoperated demonstrations based on LSTM networks with skip-connections. The use of affine transformations was proposed to map the trajectory demonstrations into a coordinate space that facilitates learning of the robot control policies. The robotic platform used for this work is introduced, which consists of the adaptation of an industrial dual-arm robot for surgical tasks based on 3D printed grippers extensions. A simulation environment for the collection of trajectory demonstrations in virtual reality was presented, as well as its use to test robot control policies.

4. EXPERIMENTS AND RESULTS

To validate the proposed framework, we conducted experiments of autonomous surgical task execution in a simulation environment. The experiments were performed for determining the feasibility of our approach to automate full dexterous surgical tasks that require the coordination of two robotic arms. We evaluate virtually the robot's capability to execute independent surgemes as well as full task execution using trained policies. The task execution experiments aim at testing an important challenge in deep imitation learning, which is the ability of generalization of the learned policies on variable initial conditions.

4.1 Description of the experimental setting

We evaluated the robot control policies for both tasks in the CoppeliaSim robotics simulator for 80 independent episodes (trials). At the beginning of each trial the robot was sent to a home position and the environment was initialized at a random state. The initial environment state was obtained from the same random distribution employed during data collection. During the peg transfer task, policy testing was performed over variations of pegboard rotation and translation as well as pegboard configurations (random peg number and direction of transfer, such as left-to-right). In the debridement task, the position of the tissue to be removed and the tray position were randomized.

For each episode of the task, the policy was run frame by frame until the task was completed or to a maximum of 160 time steps. The environment state and the robot state at each time step were stored for analysis of quantitative metrics over the generated trajectories. Per surgeme performance and overall task success were computed based on videos of experiments and recorded states of the system.

4.2 Results on task execution

Overall task success was determined programmatically based on the distance between the target object and the goal and corroborated visually over the recorded videos. The task was considered successful when the absolute distance was below the threshold ε.

$$\left|X_g - \vec{o}_p\right| < \varepsilon \tag{30}$$

Surgeme execution success was calculated as an indicator of the capacity that the networks have for learning the motion primitives of the task. First, the observed surgemes in the video recordings of the experiments were manually annotated. The trials were segmented into surgemes using a video annotation tool for surgical tasks developed for this purpose (Xingguang et al., 2019). Figure 24 shows the annotation tool utilized for the analysis. The annotation tool allows for precise annotation of beginning, end, name and success or failure.

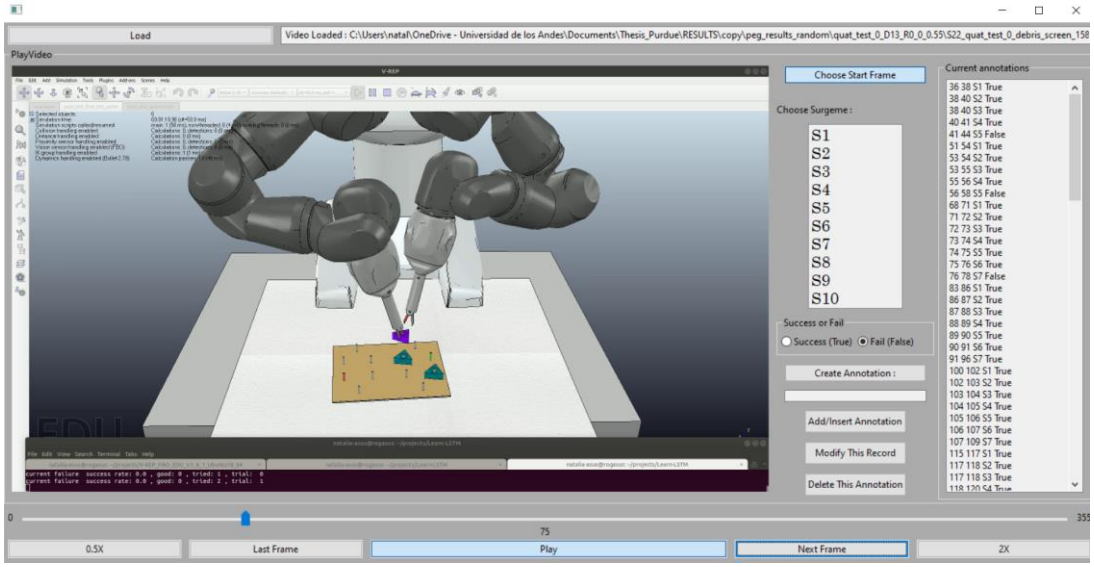

Figure 24. Tool used to annotate individual surgemes from recorded experiments.

In the following section, the surgeme success rate and overall task success for both surgical tasks are reported.

4.2.1 Peg transfer task evaluation

In this section, results for task and surgeme execution for the peg transfer task are presented. The surgeme "*approach"* referred to as the initial motion of a robotic arm towards the start peg. If the robotic arms did not move towards the start peg, then the surgeme approach was considered a failure. The surgeme *"align and grasp"* consisted of a motion to align the gripper with the desired part and closing the gripper to grasp the part. If the part was not held by the gripper after the closing

action, then the grasping surgeme failed. After grasping the part, the *"lift"* surgeme, consisted of successfully lifting the part from the peg without dropping it.

The surgeme "*transfer - get together*" consisted of the two robotic arms moving towards each other, while the surgeme *"transfer – exchange"* consisted of the opening of one of the grippers, having the other gripper grasp the part, and closing action of the first gripper, resulting in an exchange of a part between grippers. During object exchange, the task required accurate forceps alignment and synchronization between the open and closing actions to provide an accurate grasp. After the part had been handed over to the second arm, the surgeme *"approach target peg"* consisted of moving the grasped peg towards the goal. Finally, the surgeme *"align and place"* consisted of aligning and positioning the part around the goal peg. A collision with the peg or lack of assembly onto the peg would be considered a failure.

From the experiments observed, surgemes were assessed as success/failure using the evaluation process previously described. Figure 25 shows the results of this surgeme assessment. Because the task was evaluated in simulation instead of in a real setting, the success of the task could be confirmed programmatically. The overall peg transfer task was considered successful when the part was placed around the peg and the maximum distance from the part center to peg center was equal or less than 15 mm. A summary of the surgeme success rate and overall task success of our experiments is presented in Table 4.

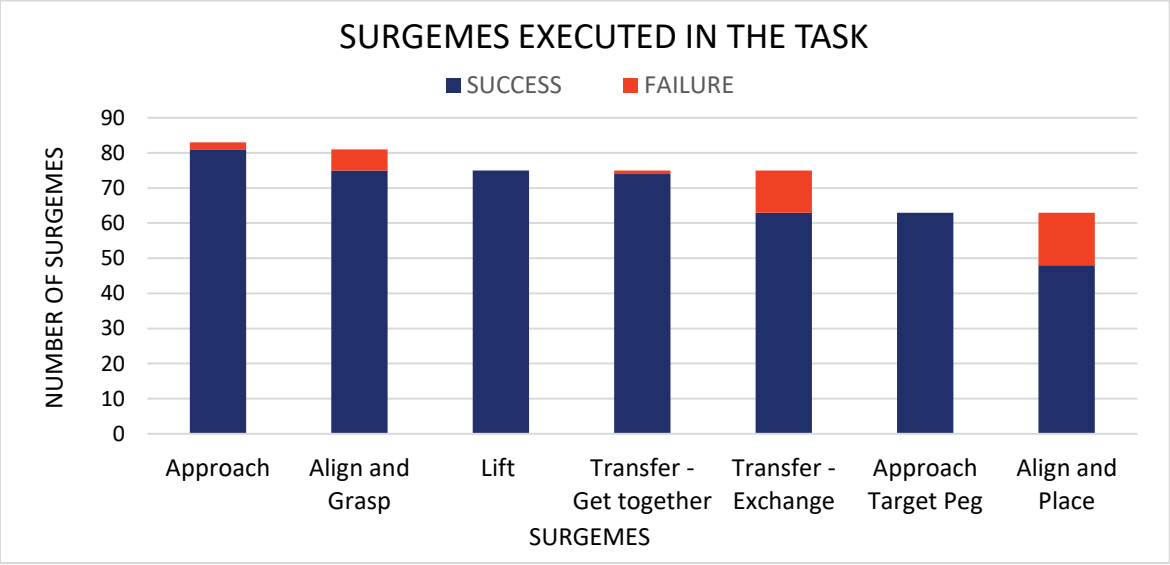

Figure 25. Successful and failed surgemes observed during task execution for the peg transfer task

Table 4. LSTM policy results for the peg transfer task.

Surgemes	Approach	Align & grasp	Lift	Transfer get together	Transfer Exchange	Approach Target Peg	Align & place	Full Task
Success	81	75	75	74	63	63	48	47
Failure	2	6	0	1	12	0	15	33
Total	83	81	75	75	75	63	63	80
Success rate	**98%**	**93%**	**100%**	**99%**	**84%**	**100%**	**76%**	**59%**

Most of the surgemes have an individual success rate over 90%, with the surgemes *"lift"* and *"approach target peg"* having a success rate of 100%. The surgemes that presented the lowest performance during task execution were the surgemes *"transfer-exchange"* with 84% success rate and the surgeme *"align and place"* with 76%. Conversely, the network policy execution results in an overall task success of 59%.

The failures observed on the *"approach"* surgeme, corresponded to the robot arms not moving towards the start peg, which resulted in unsuccessful trials. During *the "align and grasp"* surgeme, the failures were characterized by not holding the part after gripper closure. After a failure in grasping, it was observed that the robot policy often attempted to grasp again, resulting in more grasping surgemes observed than evaluated trials.

For the *"transfer - exchange"* surgeme, errors occurred when the first arm released the part before the second arm grasped it, or when the second gripper was not correctly aligned with the part before closing. For the *"align and place"* surgeme, errors were the result of poor alignment between the part and the goal peg. One potential reason for such errors is that the grasp obtained from part handover could have affected the precision of the part placement.

4.2.2 Surgical debridement task evaluation

In this section, performance metrics for surgemes and overall task success are presented for the debridement task. The surgeme "*approach"* is referred to as the initial motion of a robotic arm towards the tissue fragment to be debrided. The surgeme "*grasp"* consisted of successfully closing a gripper around the tissue fragment in a way that would be securely held by the gripper. After grasp, the *"retract"* surgeme consisted of pulling the grasped fragment upwards in order to

separate it from the remaining healthy tissue. At last, the surgeme *"cut",* consisted of closing the forceps around the tissue that was connected to the fragment.

In the simulation environment, a sensor was placed on each gripper to detect tissue located within the end-effector tooltips during the cut action. The cut action is successful when the end-effector closes around the tissue connected to the fragment. After cutting, the last surgeme*, "place on tray"* consisted of moving the excised tissue to the surgical tray.

Each surgeme observed during the experiments was annotated using the annotation tool previously described. The success and failures of each surgeme are presented in Figure 26. The overall task success for the debridement task measures whether the extracted tissue was properly positioned on the surgical tray. We considered this task successful if the tissue fragment was detached from the rest, and the tissue was properly deposited on the tray (within 4 cm or less of its center). Table 5 shows the results for surgeme success rate and overall task performance.

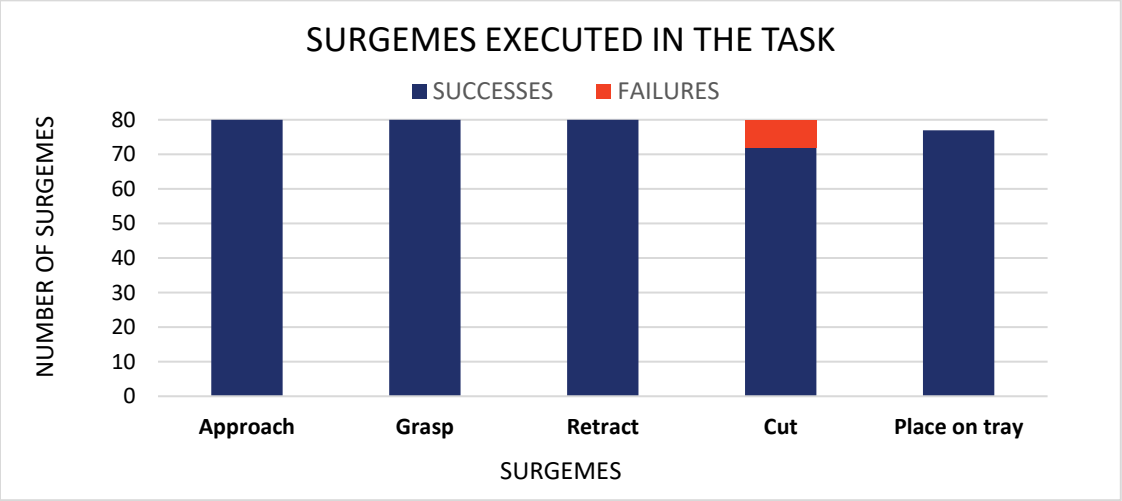

Figure 26. Results for surgeme success in task execution for the debridement task.

Table 5. LSTM policy results for the debridement task.

Surgemes	**Approach**	**Grasp**	**Retract**	**Cut**	**Place on tray**	**Full task**
Successes	80	80	80	72	77	72
Failures	0	0	0	8	0	8
Total	80	80	80	80	77	80
Success rate	**100%**	**100%**	**100%**	**90%**	**100%**	**90%**

Out of 80 trials, 72 were completed successfully while 8 were considered failures, leading to an overall task success of 90%. In terms of success rate in surgeme execution, we found that the overall success of the trial was determined by the cut surgeme. This surgeme was successful 90% of the times. The remaining surgemes “Approach”, “Grasp”, ”Retract”, and “Place on tray” were successful 100% of times.

We identified two types of failures in the cutting action, which were: failed to start the cutting action and attempted to cut at the wrong position.

4.2.3 Comparing robot performance with human demonstrations

This section presents a comparison of robot performance versus human demonstrations.

Human task success was calculated over the recorded demonstrations. The human task success metric corresponds to the number of successful demonstrations divided over the total number of demonstrations. This information could be retrieved from the data recorded sessions, as the task demonstrations errors were annotated during data collection using controller commands (refer to section ***Data collection*** for an explanation of this). Errors included collisions with the table or the peg, failing to grasp the target object, dropping the target object, or failing to cut the connecting tissue for debridement in one cutting action. The results for human versus robot task success are shown in Table 6.

Table 6. Task success comparison between human demonstrations and robot control policy.

Task success		***Human demonstrations***	***Robot***
Peg Transfer	*Successes*	839	47
	Total episodes	1065	80
	Success rate	**79%**	**59%**
Debridement	*Successes*	934	72
	Total episodes	1102	80
	Success rate	**85%**	**90%**

We can infer from these results that both tasks presented some difficulty to the human operator, as multiple errors occurred during task demonstrations. The lack of haptic feedback during robot

teleoperation in the virtual reality scenario would have been a factor that affected the performance of the human during task execution. The success rate for the human in the peg transfer task was lower than that for the surgical debridement, indicating that the peg transfer task was a more challenging task for both the human and the robot. The trained policy for the surgical debridement task shows a 5% higher success rate than the human counterpart, while the robot control policy for the peg transfer task was 20% less successful than the human during task execution.

4.2.4 Comparison of orientation representation for robot pose control

Experiments were conducted to study the impact of different orientation representations to describe tooltip pose during policy learning. Unit quaternions and Euler angles are orientation representations widely used in robotics. In this section, the impact of using unit quaternions versus employing XYZ Euler angles in task success is evaluated.

To perform this comparison, Euler angle representations for robot tooltip orientation were extracted directly from the CoppeliaSim environment during data collection. These demonstrations were stored as an alternative dataset. For each rotation representation and task, five neural networks were trained according to the procedure described in section 3.6 for a range of hidden nodes per layer (from 64-800). The effects of the two representations in policy learning were evaluated during policy execution in the simulation environment. Each network was tested during 20 trials and their overall task success was computed. The results for task success and orientation representation are summarized in Table 7 and Table 8.

Table 7. Task success rate in the peg transfer task for different orientation representations.

Number of hidden nodes per layer	***128***	***256***	***500***	***700***	***800***
Quaternion	5%	20%	50%	60%	20%
Euler angles	0%	0%	15%	30%	0%

Table 8. Task success rate in the debridement task for different orientation representations.

Number of hidden nodes per layer	***64***	***128***	***256***	***500***	***700***
Quaternion	0%	20%	90%	65%	25%
Euler angles	0%	10%	30%	30%	25%

The results indicate that the tooltip orientation representation can greatly impact the performance of the task. For both tasks, the networks that used quaternion representation displayed a higher task success rate than the networks trained using Euler angles. In the surgical debridement task, the best performing network (256 hidden nodes) using quaternion displayed 90% task success rate while the Euler network resulted in 30%. For the peg transfer task, using quaternion representation achieved 60% success rate, while the best result obtained using Euler angles was 30%. These results indicate that unit quaternions are a more adequate representation than Euler angles for robot pose control for LSTM network policy learning.

The issues in training LSTM policies using Euler angles can be caused by the possible singularities of the Euler angles representation when two of the three rotations align. This problem is also known as gimbal lock (Lepetit & Fua, 2005). Our results were consistent with previous work on pose prediction using neural networks in forward kinematics (Grassmann & Burgner-Kahrs, 2019) and 3D point cloud pose estimation (Y. Zhou et al., 2019). Such works found that higher error rates for orientation estimation are found when using Euler angles than quaternion representations. The results of our comparison indicate that quaternions are also preferable for training LSTM networks for robot task execution.

4.2.5 Unified robot coordinate space and original robot space comparison

Experiments were conducted to show the effect of using the unified robot space to represent the input dataset D'_{task} versus directly employing the original dataset D_{task}. For each task, an LSTM network was trained using the dataset in the original robot space by following the procedure described in section 3.6. The networks were tested in the simulation environment for 40 trials each. The observed surgemes were annotated and the overall task success computed.

Figure 27 and Figure 28 show a comparison between the performance obtained with the original robot space and the unified robot space. In these figures, the y-axis represents the percentage of trials for which the surgeme was executed successfully. This measurement is an indication of the level of task completion. The percentage of trials for which the last surgeme was successful is an indication of the percentage of trials for which the task was fully completed.

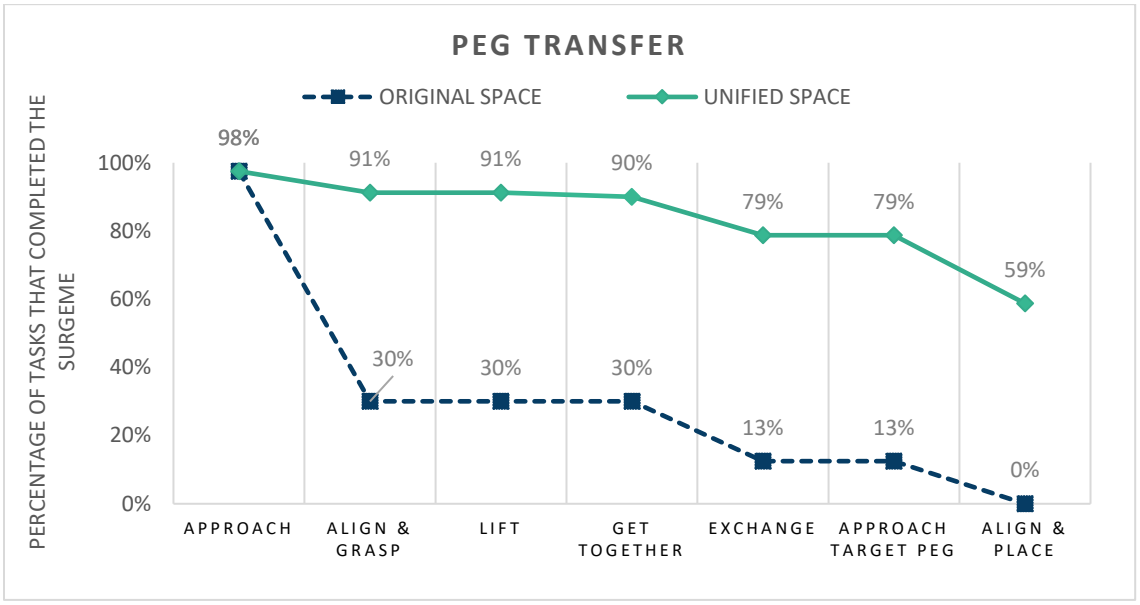

Figure 27. Percentage of trials where each surgeme was performed successfully in the peg transfer task.

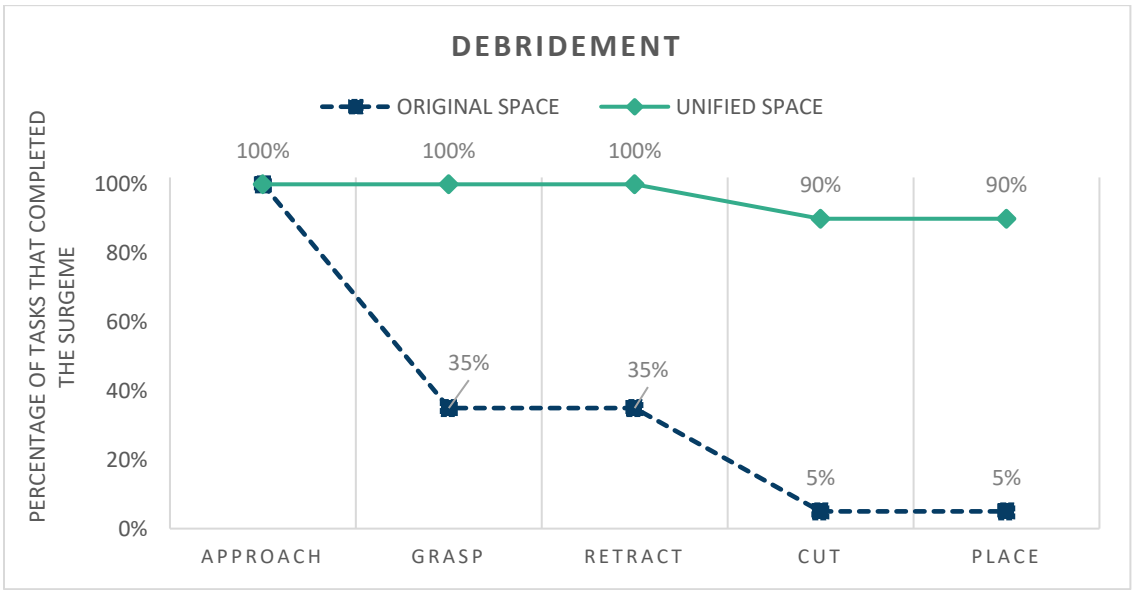

Figure 28. Percentage of trials where each surgeme was performed successfully in the debridement task.

The networks trained using the original robot space produced lower results for full task execution compared to the unified space. In the peg transfer task, none of the trials for the original robot space were observed to successfully perform all surgemes to complete the full task, which resulted in a 0% success rate. For the debridement task, only 5% of the trials successfully performed all surgemes for the full completion of the task. The use of the unified robot space over the same dataset boosted the task execution performance from 0 to 59% for the peg transfer task and from 5% to 90% for the debridement task.

Training LSTM policies directly over the original dataset was found to be ineffective to learn full task execution. For both tasks, the policies trained with the original robot space were inaccurate, resulting in the majority of the trials failing to grasp the target objects. For the peg transfer tasks, only 30% of the trials were successful at grasping the object, versus 91% for the unified robot space. 13% of the trials successfully exchanged the peg, but additional difficulties during the object placement resulted in none of the trials successfully finishing the peg transfer. For the debridement task, only 35% of the trials could grasp the object successfully, versus 100% for the unified robot space. While only 5% of the trials demonstrated a successful cut, versus 90% for the unified space. The results showcase the benefit of using the unified robot space to train more successful robot control policies with the used LSTM network architecture.

4.2.6 Discussion

The task success of 90% obtained in the surgical debridement task shows that our network can achieve good performance during task execution and generalize to randomized settings. In contrast to previous work in debridement automation, our framework can automate the full bi-manual debridement task without the need for segmenting or preprogramming primitive actions.

However, the results for the peg transfer task indicate that the approach cannot generalize equally well among tasks. Considering the high dexterity required for the peg transfer task, a 59% total execution success over a highly randomized scenario demonstrate encouraging results for its future automation. The peg transfer task is a complex task for automation because it requires precise tooltip orientation and position due to the shape of the object and the small clearance between the object and the peg. For example, the target object needs to be grasped perpendicular to one of the sides of the triangle in order to generate a good grasp. This causes the ideal grasping points

demonstrated by the human to be misaligned with the center of the object. During policy training, the network is implicitly trying to optimize for ideal grasping solutions by minimizing the mean square error over the predictions. This can potentially lead the network to produce grasping attempts that aim at the average of the demonstrated grasping points. However, the average of the demonstrated grasps in the peg transfer task might be located in the middle of the part, which would not represent a good grasp solution. Such misalignment can difficult the learning process of the peg transfer task, while for the debridement task the ideal grasps would be aligned with the center of the object.

In contrast to the debridement task, the peg transfer required to exchange the object between robot arms. Such maneuver required the object to be held by both grippers at the same time with adequate tool position and orientation. Additionally, if the first end-effector opened before the second had secured a good grasp, the object would fall. If the first end-effector does not release or is not retrieved from the object, then it can interfere with the motion of the other gripper that is holding the object.

Previous work on peg transfer automation has parametrized the primitive actions or has simplified the task by not modeling the object handover portion of the task. Our framework considers the task complexity involved in object handover for full task automation. Moreover, the approach presented allowed learning the robot pose directly from human demonstrations and does not rely on primitive action segmentation or a predefined set of motion primitives.

In contrast to the human, the robot control policy does not possess visual feedback about the quality of the grasp that it can use to correct for errors. Instead, the robot action predictions are based solely on the current position of the object. This represents an additional difficulty for the peg transfer task since the orientation of the triangular object is critical for the handover and part placement. Therefore, incorporating objects' orientation or visual feedback in the framework could potentially improve task success rate.

In non-surgical scenarios, it has been observed that increasing the number of human demonstrations results in a higher success rate for manipulation tasks. We believe that increasing the number of demonstrations and adding greater variability can improve the success rate of both tasks. Surgical teleoperated systems and surgical simulation software can potentially be employed

to acquire more demonstrations. Potentially, this can lead to higher success rates in dexterous task execution using our framework. In this work, the human demonstrations were limited to one single user. Variability could potentially be added to the dataset by collecting demonstrations from different users. In addition, strategies to inject noise in teleoperation during data collection could be explored to force the operator to correct for the errors to increase the robustness of the learned policies.

The results obtained for the debridement task show that the architecture proposed can be applied to general surgical tasks for learning dual-arm collaboration. We observe that training a policy for both arms allows the network to implicitly control one robot arm to perform the object picking operation and simultaneously control the opposite arm to perform the complementary action. This behavior indicates that our framework can learn robot arm coordination during manipulation for different surgical tasks directly from human demonstrations.

4.3 Additional performance metrics

In this section, we compare the trajectories generated by the robot control policy versus the human demonstrations, using a set of quantitative metrics. First, the learning curve of the human user demonstrations was evaluated using the CUSUM analysis, to establish the learning phases of the user in the learning curve. Then, the trajectory demonstrations of each human phase were analyzed against the trajectories obtained from the robot control policy. The assessed metrics include total movement path length, angular length, and task completion time.

4.3.1 Cumulative Sum Analysis

The CUSUM curve is the cumulative sum of the difference between each data point in the learning curve and the mean of all the data points. We created the CUSUM curve for the task completion time from human demonstrations provided by a single user during data collection. The CUSUM analysis (Liao et al., 2014) was applied to identify different phases of the learning curve in each surgical task by analyzing the peaks and trends in the learning curve.

The peaks in the curve were identified by the trial number x_i where there was a local maximum in the CUSUM curve over the interval $(x_i - \delta, x_i + \delta)$. The local maximum was identified as the trial number x_i where the following inequality holds:

$$CUSUM(x) < \mathrm{CUSUM}(x_\mathrm{i}) \; \forall \; x \in (x_i - \delta, x_i + \delta)$$

The peak points in the CUSUM curve were identified using $\delta = 50$ trials. Then, the peak prominence was calculated as the difference between the local maximum at x_i and the local minima at x_j over the interval $(x_i - \delta, x_i + \delta)$ for each identified peak. The local minima at x_j is the point where the following inequality holds:

$$CUSUM(x) > \mathrm{CUSUM}(x_j) \; \forall \; x \in (x_i - \delta, x_i + \delta)$$

Thus, the peak prominence was defined as $\mathrm{CUSUM}(x_\mathrm{i}) - \mathrm{CUSUM}(x_j)$.

In the peg transfer task, exactly 2 local maximum points were found on the learning curve using $\delta = 50$. By analyzing the trend of the curve around these peaks, 3 phases of the learning curve could be identified. The Phase I or "novice phase" was observed from the beginning of the learning curve to the first peak at trial 287, characterized by a positive slope. The Phase II or "intermediate phase" was identified between the trials 288 and 510 which showed a change in the curve slope indicating accumulation of expertise. The Phase III or "expert" phase was identified from trials 511 onward, which demonstrated a strong negative slope indicating mastery in the task. The CUSUM curve and the three identified phases are shown in Figure 29.

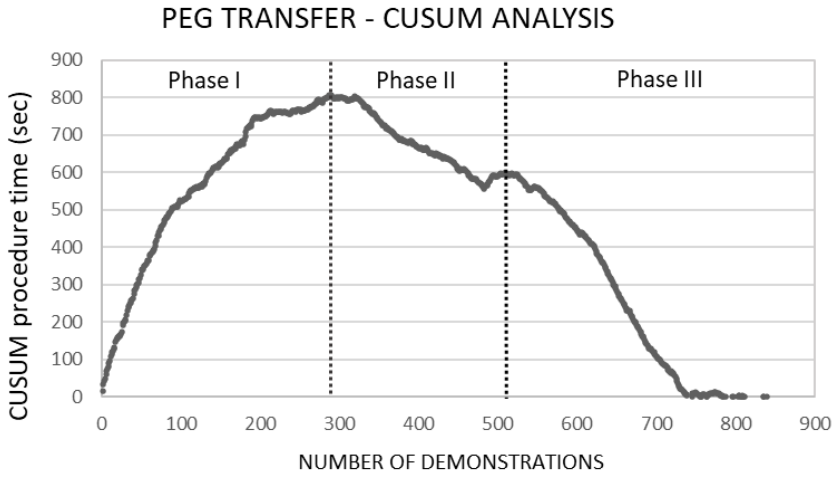

Figure 29. Cumulative sum analysis for the demonstrations in the peg transfer task

For the surgical debridement task, three phases in the curve were identified between the two most prominent peaks. The Phase I or "novice" phase demonstrated a strong rising pattern from the start of the data collection to the first major peak at trial 386. The Phase II or "intermediate" phase was identified between the trials 387 and 581, which showed a fluctuating pattern until the second major peak. The Phase II or "expert" phase was identified after trial 581 and the CUSUM value was characterized by a strong decreasing trend which indicates gained expertise. The cumulative sum analysis and identified phases for the learning curve are shown in Figure 30

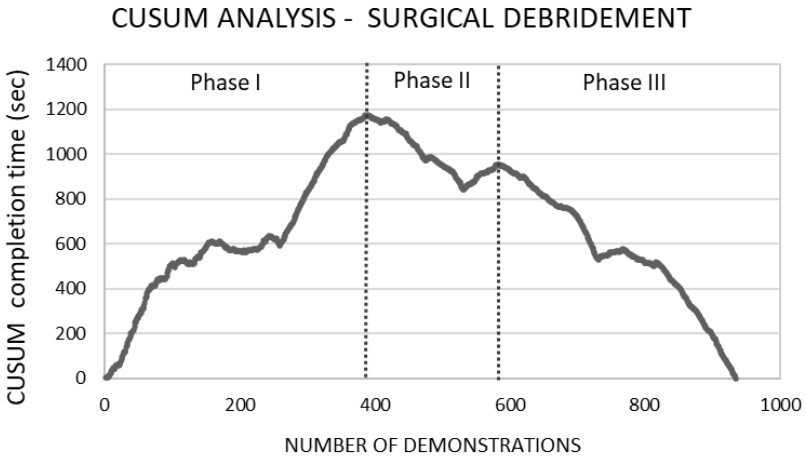

Figure 30. Cumulative sum analysis for the demonstrations in the surgical debridement task

Using the CUSUM analysis, 3 phases were identified, for each surgical task which can be used to assess quantitative metrics of movement over the trajectory demonstrations during different phases.

4.3.2 Quantitative metrics of movement

Three different metrics of movement were evaluated for comparison between the robot generated trajectories and three different learning phases of the human demonstrator. We compared two time-independent metrics for the characterization of robot tooltip motion: the path length and the angular length. In addition to the time-independent metrics, the completion time of the recorded human demonstrations and the robot policy executions were recorded. We consider only the trials that were successful during execution (for both the human demonstrator and the robot), as they represent complete trajectories for the execution of the task. For the human demonstrations, the trajectories automatically annotated during data collection are used, from the first movement of

the robot until the end of the task marked by the human. For the robot control policy, the kinematic trajectories used are those recorded during task execution from the first command sent until task completion.

Path length: The path length was computed as the length of the curve shown by the robotic tooltips during task performance (in m). This was calculated as the sum of the Euclidean distance between consecutive tooltip positions in the recorded trajectories. For each episode of the task, the total path length is reported, which is sum of the path length of the left and right tooltips.

Angular length: The angular length refers to the cumulative changes in the rotation of the robot tooltips during task execution. The angular length was calculated based on the recorded quaternions as the sum of the geodesic distance between the quaternion of the previous and the current time step. This distance was calculated as the angle ∅ between subsequent quaternions (q_i, q_{i+1}) which can be retrieved from the logarithmic mapping between them (Solà, 2017):

$$\emptyset_i = |Log({q_i}^{-1} q_{i+1})| \tag{31}$$

The angular length is calculated as the cumulative sum of ∅ over all consecutive pairs of points in the recorded trajectories.

$$AL = \sum_{i=0}^{i=N-1} |Log({q_i}^{-1} q_{i+1})| \tag{32}$$

We evaluate the total angular length as the sum of the angular length of the left and right tooltips.

Completion time: It was calculated as the total time for task completion. It is computed from the start of the robot tooltip motion until the end of the annotated trajectory.

Results for quantitative metrics of movement

Path length

The mean path length for the peg transfer task and the surgical debridement are shown in Figure 31 and Figure 32 and respectively. A one-way ANOVA followed by a post hoc Tukey-Kramer procedure was applied to examine the significant differences for path length metrics between the three human learning phases and the robot control policy.

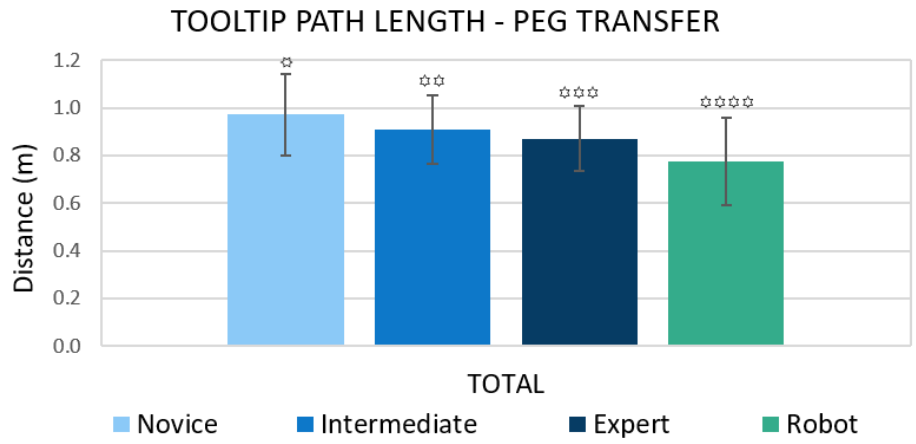

Figure 31. Path length results for the peg transfer task. Groups that do not share a symbol are significantly different $p<0.05$.

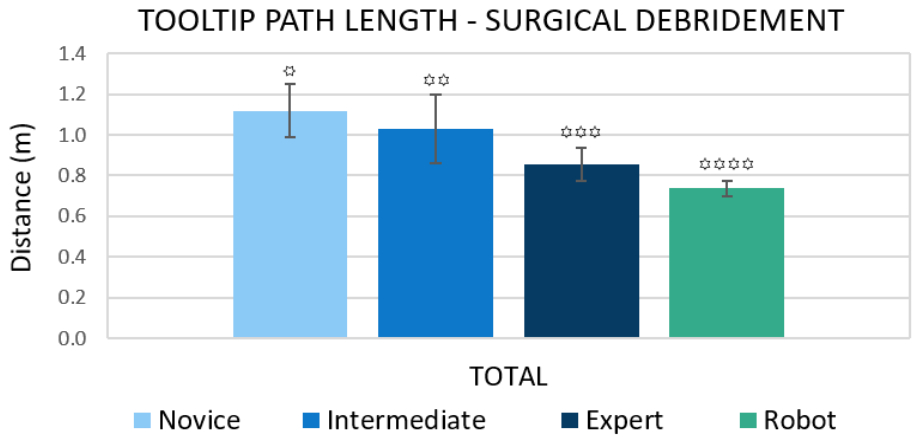

Figure 32. Path length results for the surgical debridement task. Groups that do not share a symbol are significantly different $p<0.05$.

The results for the debridement and the peg transfer task demonstrate similar trends in total path length comparison. For both tasks, the total path length between all the human phases and the robot are significantly different ($p<0.05$). The results of total path length between the human phases show that the path lengths are reduced as proficiency increases. By comparing the human with the robot, we found that the robot total path length was significantly shorter than all the human phases ($p<0.001$) for both tasks. These results indicate that the tooltip path length during robot execution displays a better economy of motion than the human demonstrations.

Angular length

To analyze the changes in tool orientation between the human phases and the robot, the average total angular length for each surgical task was calculated. To analyze the robot control policy with respect to the human demonstrations, a one-way ANOVA and a Tukey-Kramer pairwise comparison for each one of the angular length metrics and tasks was conducted. The results for the angular length are reported in Figure 33 and Figure 34.

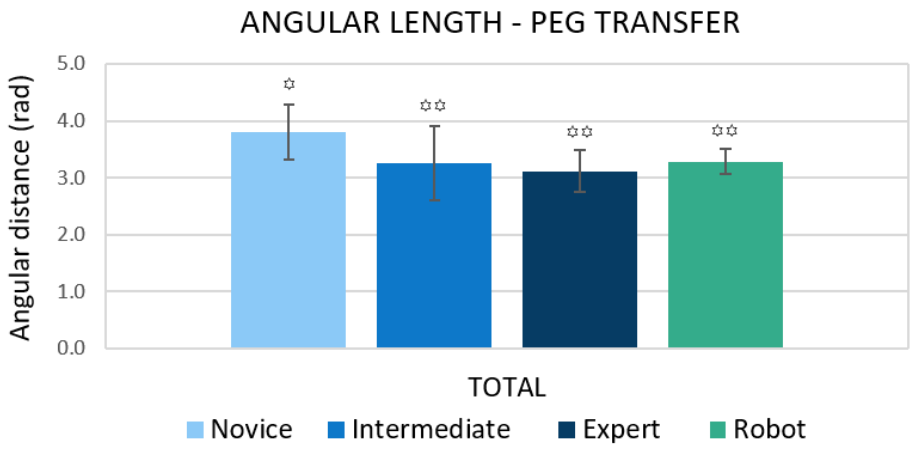

Figure 33. Angular length results for the peg transfer task. Groups that do not share a symbol are significantly different $p<0.05$

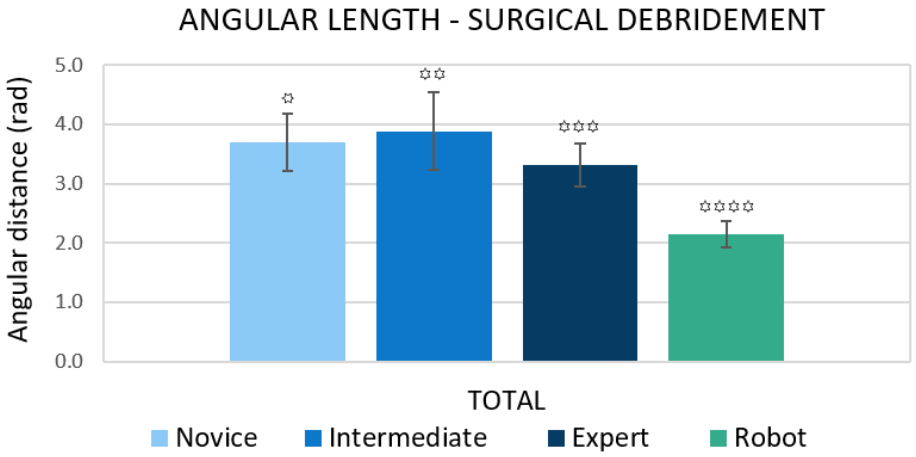

Figure 34. Angular length results for the surgical debridement task. Groups that do not share a symbol are significantly different $p<0.05$.

In the peg transfer task, the angular length results show that the intermediate and expert have significantly less tool re-orientations than the novice ($p<0.001$**)**. Similarly, the robot policy displays a significantly shorter angular length over the novice group ($p<0.001$). There is no significant difference between the angular length of the robot and the intermediate and expert user. These results indicate that the angular motions for the robot control policy display a better motion economy than the human novice and are similar in angular changes to the intermediate and expert.

In the surgical debridement task, the expert displays a significantly shorter angular path than the intermediate and the novice ($p<0.001$). Thus, indicating than the expert human requires less tool reorientations to perform the task. The robot displays a significantly shorter angular path than all human phases ($p<0.001$), which shows that the robot can perform the task with less changes in tool orientation.

While the path length indicates the translation of the tooltip motion, the angular length is an indication of how much the tool is reoriented during task execution. The results showed that experts on average display less tool reorientations throughout the task, which indicates that experts present more efficient motions in terms of tool orientation changes. A possible explanation for this pattern is that motions that represent orientation corrections during teleoperation can result in an increased angular length.

Completion time

The results of mean completion time for the surgical debridement and the peg transfer task are presented in Figure 36 and Figure 35 respectively. A one-way ANOVA was performed over the three human learning phases and the robot control policy, followed by a Tukey-Kramer pairwise comparison. The *"Robot"* completion time corresponds to the time from the start of the task to its successful completion during policy testing as described in the section 3.7. Such online policy testing relied on the communication between the python API and the environment in CoppeliaSim.

The *"robot theoretical"* completion time, corresponds to the minimum time the robot required to execute the recorded trajectories in a sequence, by eliminating robot idle time caused by waiting for the next robot command through the python API. This "theoretical robot completion time" is calculated by loading the recorded trajectories during policy execution in the CoppeliaSim

environment and commanding the robot to follow such trajectories point by point. During this test, a point in the trajectory was considered achieved when the end-effector reached a position within 0.5 mm of the target position and an orientation within 1 degree of the commanded orientation. Thus, the theoretical robot completion time corresponded to the minimum time the robot took to execute the given trajectories considering the YuMi robot maximum tooltip velocity of 1.5 m/s (ABB, 2019). In addition, the average computational time for robot output prediction using the LSTM network is added to the theoretical completion time for each point in the trajectory. The average computational time for robot output prediction is ($\mu = 30.7\, ms, \sigma = 1.3$) for the peg transfer and ($\mu = 29.7\, ms, \sigma = 1.5$) for the debridement task using the Nvidia GPU GeForce GTX 1070 8GB.

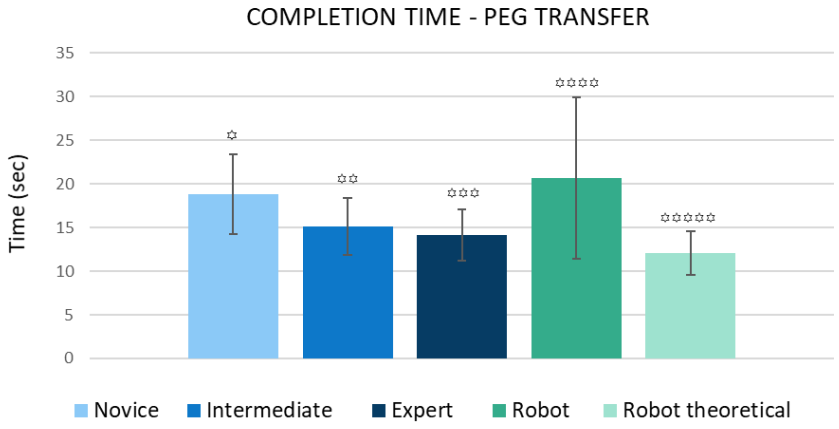

Figure 35. Completion time for the peg transfer task. Groups that do not share a symbol are significantly different p<0.05.

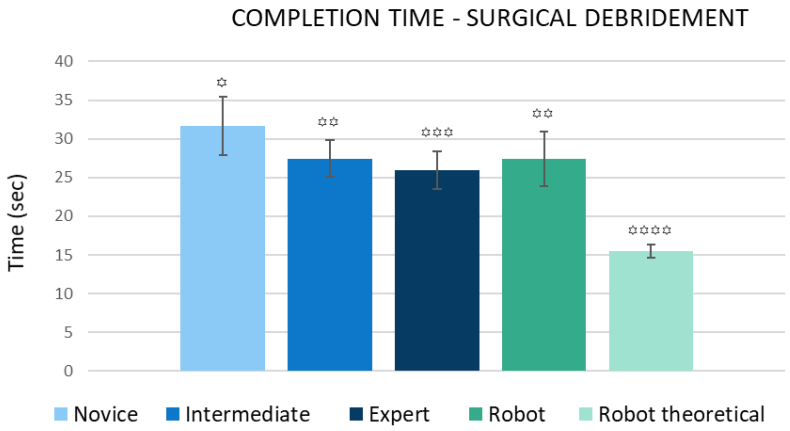

Figure 36. Completion time for the surgical debridement task. Groups that do not share a symbol are significantly different p<0.05.

The completion time for the peg transfer task shows statistical significance between all groups. The completion time for all human phases decreases as the expertise increases, with 18.8 sec for the novice, 15.1 for the "intermediate" and 14.2 for the "expert". The task completion time for the robot in the peg transfer task is significantly higher than all human bases (20.6 s), which is 1.8 seconds longer than the average novice demonstration. This result can be explained by the difference between the sampling rate of the features from the human demonstrations during network training and the feature sampling rate during execution. The policy execution sampling rate was 4Hz during all validation and testing. In contrast, the human demonstrations were subsampled at 5Hz for network training, which was selected as a hyperparameter during validation. Since the robot control policy is evaluated at 4Hz and the human data used for training was sampled at 5Hz, the control policy was tested at x0.8 times the sample rate of the human demonstrations employed for training. In contrast, the robot theoretical completion time for the peg transfer task was 12.1s (10s without the computational prediction time), which indicates that the autonomous execution of the peg transfer could potentially be reduced up to nearly half the completion time.

The results for the debridement task show that the robot mean completion time is significantly faster than the human novice (31.62 s, $p<0.001$) and significantly faster than the expert (25.9 s, $p=0.03$). The mean completion time between the robot (27.46 s) and the intermediate phase (27.46 s) were not significantly different ($p=1$). These results indicate that the trained robot policy for the debridement task was faster than the human novice demonstrations and slower than the human expert demonstrations. In addition, the theoretical robot completion time for the debridement task was 15.5s (12.6s without computational prediction time) which indicates that the task completion time could be further reduced to be less than the completion time in all human groups.

4.3.3 Discussion

The analysis performed over the total path length showed that the robot control policy overall produced shorter trajectories than the human demonstrations for task completion. These results indicate that our framework can learn paths for task execution with good motion economy. The analysis performed over the angular path showed that the robot control policies for the debridement task required less reorientations than the human demonstrations. In contrast, for the peg transfer

task, the tool orientation changes were not significantly different from the intermediate and mastery phases. These results indicate that our learning framework produces trajectories with less or similar overall orientation changes that the human demonstrations.

The completion time for the debridement task showed comparable completion times with the human, while the results for the peg transfer task showed a completion time of 1.8 seconds longer than the novice human. For future applications, the robot completion time for either task could be reduced by evaluating the robot control policy at a higher rate. Specifically, the next robot pose should be predicted as soon as the robot end-effectors reach the previously sent command in order reduce robot idle time between commands.

4.4 Summary

In this chapter, we presented the results for autonomous task execution for the peg transfer and the surgical debridement tasks using our framework. The surgeme success was analyzed to evaluate the capacity of the trained networks to learn different steps of the task. We calculated quantitative metrics of robot motion for the robot generated trajectories and compared them with human demonstrations. The task time completion was similar to the human for the debridement task and longer for the peg transfer task. Future work should focus on integrating the control policy with a real robotic system and optimizing the frequency of policy evaluation in such platform to reduce the task completion time. Our results indicate that our framework can learn dual-arm interaction for multi-step tasks such as the debridement. For the peg transfer task, the trained network displayed moderate performance and has not yet achieved the dexterity and precision necessary to reliably solve such a task. However, our work represents a step towards investigating intelligent algorithms for full task automation in dual-arm manipulation.

5. CONCLUSIONS AND FUTURE WORK

In this thesis, a framework for full task automation was presented as a holistic approach for learning two surgical tasks. An architecture for learning robot control policies using recurrent neural networks was proposed to execute the tasks while implicitly modeling dual-arm interaction. The proposed recurrent neural network takes, as an input, the state of the robot arms and the state of the environment. It continuously predicts the robot's next arms poses throughout the duration of the task. Thus, avoiding the need for prior data segmentation, using customized models for subtasks, or the design high-level controllers for module coordination. Instead, the presented architecture can be trained over the full dataset of collected examples, making it suitable for the learning of new surgical tasks. This work represents a step towards the creation of smart algorithms that leverage artificial intelligence to learn the task directly from human demonstration. The presented approach aims at achieving full task automation without the need for task expert knowledge. This, in turn, could help decrease the time and manpower required to develop algorithms for autonomous surgical task execution.

The presented approach generates policies for continuous robot control. Robot controllers that can continuously predict the next robot action based on the previous observations could have potential applications in semi-autonomous scenarios. For critical cases during teleoperation, when the user experiences loss of signal or communication delays, a continuous control policy could evaluate the stream of data that was last available by the human and automatically take over the task wherever the surgeon left off. Such capabilities could contribute to a natural switch between teleoperation and semi-automation in diverse scenarios.

The framework presented in this work is formulated as a general approach for learning distinct tasks, by leveraging task-related features, such as target object start and goal. The generality of the approach was tested in two surgical tasks. The results for the debridement task showed that the proposed framework allows to learn robot control policies for full task execution that implicitly learn dual-arm interaction. The results for the peg transfer task showed that the control policies can also model the interaction between the arms for different tasks. In addition, the framework generates trajectories for task execution with similar or less changes in position and orientation of

the tooltips. Which demonstrated that the policy learns trajectories with good motion economy. Producing less changes in tool position and orientation can be beneficial during task execution to avoid repositioning of the tool through the procedure. Moreover, the results for testing the policies under random initial conditions indicate that the framework can generalize well to variable settings for tasks such as debridement.

As the formulation is not limited to the selected tasks, it can be formulated for general robotic applications such as dual-arm object manipulation for industrial settings or household robotics. The proposed approach can potentially be used for an ample assortment of manipulation tasks, as it models the control of end-effector position, orientation as well as gripper opening and closing actions. For example, the framework could be formulated to perform a dual-arm robotic furniture assembly.

To enable the use of deep learning approaches from demonstrations, we introduce the use of a coordinate space representation that captures commonalities between the recorded examples to encompass multiple variations from those examples in the original setting. We called this representation the unified robot coordinate space. By training the network using such representation, the knowledge of task variation can be exploited directly during network training. This allows to learn control policies in a more sample efficient manner since one example seen by the network conveys multiple variations of examples in the original robot setting. For instance, for the peg transfer case, a demonstration in the unified space conveys multiple rotations and translations of the beg board on the working surface of the robot.

The integration of such representation to the learning pipeline allows learning control policies for complex tasks and enable its use in scenarios with variable initial conditions of the environment. Note that, the unified robot coordinate space is a technique to represent the dataset, which could be integrated into neural network architectures different than ours and applied to general manipulation tasks.

5.1 Limitations

The framework for surgical task execution was validated in a simulated setting. To transfer the robot control policies to a real robot setting, an autonomous vision system should be integrated to

capture the state of the environment. As simulation environments can differ from real settings, the policies trained in simulation might not be directly transferable to a real platform. For example, the framework assumes that the tissue to be removed in the debridement task is rigid. However, real tissue manipulation is subject to deformations that are not accounted for in simulation. Thus, better and more realistic simulation environments may be required to evaluate appropriate strategies to transfer policies trained in simulation to a real setting.

Currently, the implementation of the algorithm in CoppeliaSim uses the python API. The python API suffers from latencies in the socket communication between python and the simulation environment, as well as delays between the main thread and other communication threads (James et al., 2019). Such latencies currently limit the communication rate between python and the robot simulated in CoppeliaSim. Future attempts should involve implementing the same framework using the PyRep toolkit (James et al., 2019) in order to reduce such delays and test robot control policies at higher communication rates.

Currently, our framework is based on robot pose prediction and does not explicitly consider robot joint configuration. Thus, the method requires to employ a separate module that solves the inverse kinematics of the robot. For this reason, it is possible that some pose commands predicted by the network would produce strong changes in the robot configuration. Moreover, to achieve feasible robot commands, it is desirable to constrain the position of the relevant objects in the task to a subset of the robot workspace where the robot can comfortably operate.

The framework is formulated for general dual-arm robotic systems. However, it was evaluated using only one robotic platform. Future work should focus on testing the validity of the framework in different surgical robotic systems such as the da Vinci Research Kit or the Taurus robot which are available at the lab. The Taurus robot has limited joint range and workspace compared to the YuMi robot. Thus, some commands predicted by the policy might not be reachable by the system or could potentially lead to robot joint configurations that are unsafe. In addition, our framework assumes that robot pose estimation is accurate. Implementing the policy in cable-driven systems such as the Raven II and the da Vinci research kit would require adding calibration routines to reduce possible inaccuracies in robot pose estimation (Hwang et al., 2020).

Only one subject was used to gather demonstrations for policy training. In order to make our approach generalizable, several users should execute the surgical tasks, and the models need to be trained with at least 10 users. Data coming from multiple users will add variations that could potentially increase performance and add robustness to the network. In our analysis, the robot control policies were compared with the demonstrations of one human that were classified in 3 levels of expertise. The recruitment of new users for data collection should include those expertise levels (novice, intermediate, and expert).

5.2 Future work

Future work can focus on transferring robot control policies between domains. The control policies in our framework predict the required pose and gripper state for the robots' end-effectors. Thus, policies trained on one platform could potentially be tested on a second system by commanding the robot to follow the predicted tooltip poses using inverse kinematics. Such policies might be transferable if the surgical tools are the same and the workspace of the second robotic system is similar to the workspace in the original system.

Currently, the framework does not account for target object orientation. However, tasks such as suturing and needle insertion could heavily rely on modeling needle orientation for proper handling. Therefore, an extension of our approach to such tasks should consider adding additional features in the state space to convey the orientation of the target object. For example, the object orientation could be predicted with a vision module and added to the state space of the network.

Future work will focus on extending the proposed framework for surgical tasks that possess more complex states. For example, the pattern cutting task cannot easily be modeled based on the position of one target object. Instead, it might be necessary to convey the desired pattern to be cut, the tension of the tissue, as well as to monitor the process of the cutting procedure. Incorporating visual feedback to the framework could be evaluated as an option for conveying more comprehensive states of the system that could enable modeling complex tasks.

5.3 Research questions

Three research questions were posed at the beginning of this thesis and were addressed throughout this thesis. The first question concerned the formulation of a learning approach to learn multi-step tasks in the context of surgical robotics. The second question was about the integration of dual-arm manipulation to deep imitation learning. The third question encompasses the metrics and experiments to be conducted to validate the developed framework for surgical task execution.

Research question 1 (RQ1): How can deep imitation learning be used to learn multi-step tasks in the context of surgical robotics?

To answer this question, a framework for learning surgical tasks from demonstrations was proposed. More specifically, a set of transformations were embedded in the learning pipeline to convey multiple variations of the original demonstrations during training. Such representation enabled training deep imitation learning policies for surgical tasks. The approach can be extended to additional tasks by directly applying the learning framework on the dataset of a new task. The framework proposed aims at modeling the full range of motions required in surgical tasks, which involve tooltip position, orientation, and gripper actions. Note that the method does not rely on the definition of domain-specific models for each task, therefore it could be formulated for an ample range of applications without the need for experts from the task domain. The application of the framework to general tasks was tested by its formulation and implementation in two distinct surgical tasks.

Research question 2 (RQ2): How to integrate dual-arm task learning for deep imitation learning?

To answer this question, a recurrent neural network architecture was formulated to jointly predict the control commands for two robotic arms as if they were part of the same system. More specifically, the state of the robot arms and their control commands were concatenated in the inputs and outputs of the recurrent neural network policy used for task execution. The policies produced collaborative behavior during task execution where both robot arms performed complementary actions necessary to fulfill the tasks.

Research question 3 (RQ3): How to evaluate the success of the learned framework?

RQ3 refers to the evaluation of the proposed network for surgical task execution. To answer this question, experiments were conducted in a simulated setting to evaluate the performance of the learned networks for the execution of two surgical tasks (peg transfer and debridement). The overall task success, the success of primitive actions, and a set of quantitative metrics were evaluated and compared with the human demonstrations. The trained networks showed favorable results in automating dual-arm surgical tasks under variable task conditions for the debridement task only. The debridement task displayed a high success rate (90%) comparable with the performance of the human (85%) while the peg transfer task showed results of (59%) which are (20%) lower than the human (79%). Our experiments indicate that the proposed framework has potential to model dual-arm collaboration. The quantitative metrics over the robot generated trajectories indicate that the robot showed similar or shorter path trajectory in both translational and rotational changes. This indicates that the robot policy is learning efficient trajectories from the demonstrations.

REFERENCES

ABB. (2019). *YuMi® IRB 14000, Data sheet, PDF*. https://search.abb.com/library/Download.aspx?DocumentID=9AKK106354A3254&LanguageCode=en&DocumentPartId=&Action=Launch

ABB. (2020). *ABB's Collaborative Robot*. https://new.abb.com/products/robotics/industrial-robots/irb-14000-yumi

Abbeel, P., Coates, A., & Ng, A. Y. (2010). Autonomous Helicopter Aerobatics through Apprenticeship Learning. *The International Journal of Robotics Research*, *29*(13), 1608–1639. https://doi.org/10.1177/0278364910371999

Abdou, M., Kamal, H., El-Tantawy, S., Abdelkhalek, A., Adel, O., Hamdy, K., & Abaas, M. (2019). End-to-End Deep Conditional Imitation Learning for Autonomous Driving. *2019 31st International Conference on Microelectronics (ICM)*, 346–350. https://doi.org/10.1109/ICM48031.2019.9021288

Abiri, A., Pensa, J., Tao, A., Ma, J., Juo, Y.-Y., Askari, S. J., Bisley, J., Rosen, J., Dutson, E. P., & Grundfest, W. S. (2019). Multi-Modal Haptic Feedback for Grip Force Reduction in Robotic Surgery. *Scientific Reports*, *9*(1), 5016. https://doi.org/10.1038/s41598-019-40821-1

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., & Savarese, S. (2016). *Social LSTM: Human Trajectory Prediction in Crowded Spaces*. 961–971. http://openaccess.thecvf.com/content_cvpr_2016/html/Alahi_Social_LSTM_Human_CVPR_2016_paper.html

Aroussi, S., & Mellouk, A. (2014). Survey on machine learning-based QoE-QoS correlation models. *2014 International Conference on Computing, Management and Telecommunications*, 200–204. https://doi.org/10.1109/ComManTel.2014.6825604

Asfour, T., Gyarfas, F., Azad, P., & Dillmann, R. (2006). Imitation Learning of Dual-Arm Manipulation Tasks in Humanoid Robots. *2006 6th IEEE-RAS International Conference on Humanoid Robots*, 40–47. https://doi.org/10.1109/ICHR.2006.321361

Aşık, O., Görer, B., & Akın, H. L. (2018). End-to-End Deep Imitation Learning: Robot Soccer Case Study. *ArXiv E-Prints*, *1807*, arXiv:1807.09205.

Atkeson, C. G., & Schaal, S. (1997, January 1). *Robot Learning From Demonstration*. ICML. https://openreview.net/forum?id=BJ4-72WuZS

Bark, K., Gomez, E. D., Rivera, C., McMahan, W., Remington, A., Murayama, K., Lee, D. I., Dumon, K., Williams, N., & Kuchenbecker, K. J. (2012). Surgical Instrument Vibrations are a Construct-Valid Measure of Technical Skill in Robotic Peg Transfer and Suturing Tasks. *Proc.\ Hamlyn Symposium on Medical Robotics*, 50–51.

Béjar Haro, B., Zappella, L., & Vidal, R. (2012). Surgical Gesture Classification from Video Data. In N. Ayache, H. Delingette, P. Golland, & K. Mori (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012* (pp. 34–41). Springer Berlin Heidelberg.

Berg, J. van den, Miller, S., Duckworth, D., Hu, H., Wan, A., Fu, X., Goldberg, K., & Abbeel, P. (2010). Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. *2010 IEEE International Conference on Robotics and Automation*, 2074–2081. https://doi.org/10.1109/ROBOT.2010.5509621

Bogaerts, B., Sels, S., Vanlanduit, S., & Penne, R. (2019). Enabling Humans to Plan Inspection Paths Using a Virtual Reality Interface. *ArXiv:1909.06077 [Cs]*. http://arxiv.org/abs/1909.06077

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., & Zieba, K. (2016). End to End Learning for Self-Driving Cars. *ArXiv:1604.07316 [Cs]*. http://arxiv.org/abs/1604.07316

Bokhari, M. B., Patel, C. B., Ramos-Valadez, D. I., Ragupathi, M., & Haas, E. M. (2011). Learning curve for robotic-assisted laparoscopic colorectal surgery. *Surgical Endoscopy*, *25*(3), 855–860. https://doi.org/10.1007/s00464-010-1281-x

Bolzoni Villaret, A., Doglietto, F., Carobbio, A., Schreiber, A., Panni, C., Piantoni, E., Guida, G., Fontanella, M. M., Nicolai, P., & Cassinis, R. (2017). Robotic Transnasal Endoscopic Skull Base Surgery: Systematic Review of the Literature and Report of a Novel Prototype for a Hybrid System (Brescia Endoscope Assistant Robotic Holder). *World Neurosurgery*, *105*(Supplement C), 875–883. https://doi.org/10.1016/j.wneu.2017.06.089

Bonaci, T., Herron, J., Yusuf, T., Yan, J., Kohno, T., & Chizeck, H. J. (2015). *To Make a Robot Secure: An Experimental Analysis of Cyber Security Threats Against Teleoperated Surgical Robotics* *. https://arxiv.org/pdf/1504.04339.pdf

Brown, J. D., O'Brien, C. E., Leung, S. C., Dumon, K. R., Lee, D. I., & Kuchenbecker, K. J. (2017). Using Contact Forces and Robot Arm Accelerations to Automatically Rate Surgeon Skill at Peg Transfer. *IEEE Transactions on Biomedical Engineering*, *64*(9), 2263–2275. https://doi.org/10.1109/TBME.2016.2634861

Brownlee, J. (2017). *Long Short-Term Memory Networks With Python: Develop Sequence Prediction Models with Deep Learning*. Machine Learning Mastery.

Burton, S. (2010). Coping with the Curse of Dimensionality by Combining Linear Programming and Reinforcement Learning. *All Graduate Theses and Dissertations*. https://digitalcommons.usu.edu/etd/559

Capolei, M. C., Haiyan Wu, Andersen, N. A., & Ravn, O. (2017). Positioning the laparoscopic camera with industrial robot arm. *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*, 138–143. https://doi.org/10.1109/ICCAR.2017.7942675

Chandrasekaran, K., Sathuluri, A., & Thondiyath, A. (2017). MagNex—Expendable robotic surgical tooltip. *Proceedings - IEEE International Conference on Robotics and Automation*, 4221–4226. https://doi.org/10.1109/ICRA.2017.7989486

Chernova, S., & Thomaz, A. L. (2014). Robot Learning from Human Teachers. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *8*(3), 1–121. https://doi.org/10.2200/S00568ED1V01Y201402AIM028

Clinkard, D., Holden, M., Ungi, T., Messenger, D., Davison, C., Fichtinger, G., & McGraw, R. (2015). The Development and Validation of Hand Motion Analysis to Evaluate Competency in Central Line Catheterization. *Academic Emergency Medicine*, *22*(2), 212–218. https://doi.org/10.1111/acem.12590

CoppeliaRobotics. (2020a). *Remote API functions (Python)*. https://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsPython.htm

CoppeliaRobotics. (2020b). *Robot simulator CoppeliaSim: Create, compose, simulate, any robot. - Coppelia Robotics*. https://www.coppeliarobotics.com/

Da Vinci Surgery. (2020). http://www.davincisurgery.com/

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference On*, 248–255. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5206848

Diamantaras, K., Duch, W., & Iliadis, L. S. (2010). *Artificial Neural Networks - ICANN 2010: 20th International Conference, Thessaloniki, Greece, Septmeber 15-18, 2020, Proceedings*. Springer Science & Business Media.

DiPietro, R., Lea, C., Malpani, A., Ahmidi, N., Vedula, S. S., Lee, G. I., Lee, M. R., & Hager, G. D. (2016). Recognizing Surgical Activities with Recurrent Neural Networks. In S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, & W. Wells (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016* (pp. 551–558). Springer International Publishing.

Dolph, E., Krause, C., & Oleynikov, D. (2019). Future Robotic Systems: Microrobotics and Autonomous Robots. In S. Tsuda & O. Y. Kudsi (Eds.), *Robotic-Assisted Minimally Invasive Surgery: A Comprehensive Textbook* (pp. 329–335). Springer International Publishing. https://doi.org/10.1007/978-3-319-96866-7_40

D'Souza, M., Gendreau, J., Feng, A., Kim, L. H., Ho, A. L., & Veeravagu, A. (2019). Robotic-Assisted Spine Surgery: History, Efficacy, Cost, And Future Trends. *Robotic Surgery: Research and Reviews*, *6*, 9–23. https://doi.org/10.2147/RSRR.S190720

Du, K.-L., & Swamy, M. N. S. (2019). Reinforcement Learning. In K.-L. Du & M. N. S. Swamy (Eds.), *Neural Networks and Statistical Learning* (pp. 503–523). Springer. https://doi.org/10.1007/978-1-4471-7452-3_17

Duteille, F., & Perrot, P. (2012). Management of 2nd-degree facial burns using the Versajet® hydrosurgery system and xenograft: A prospective evaluation of 20 cases. *Burns*, *38*(5), 724–729. https://doi.org/10.1016/j.burns.2011.12.008

Ericson, L. (2018). *Flying High: Deep Imitation Learning of Optimal Control for Unmanned Aerial Vehicles*. http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-233326

Ezziyyani, M. (2020). *Advanced Intelligent Systems for Sustainable Development (AI2SD'2019): Volume 2 - Advanced Intelligent Systems for Sustainable Development Applied to Agriculture and Health*. Springer Nature.

Fard, M. J., Ameri, S., Chinnam, R. B., & Ellis, R. D. (2017). Soft Boundary Approach for Unsupervised Gesture Segmentation in Robotic-Assisted Surgery. *IEEE Robotics and Automation Letters*, *2*(1), 171–178. https://doi.org/10.1109/LRA.2016.2585303

Finn, C., & Levine, S. (2016). Deep Visual Foresight for Planning Robot Motion. *ArXiv:1610.00696 [Cs]*. http://arxiv.org/abs/1610.00696

Gao, Y., Vedula, S. S., Reiley, C. E., Ahmidi, N., Varadarajan, B., Lin, H. C., Tao, L., Zappella, L., Béjar, B., & Yuh, D. D. (2014). JHU-ISI gesture and skill assessment working set (JIGSAWS): A surgical activity dataset for human motion modeling. *MICCAI Workshop: M2CAI*, *3*, 3.

Garcia, P., Rosen, J., Kapoor, C., Noakes, M., Elbert, G., Treat, M., Ganous, T., Hanson, M., Manak, J., Hasser, C., Rohler, D., & Satava, R. (2009). Trauma pod: A semi-automated telerobotic surgical system. *International Journal of Medical Robotics and Computer Assisted Surgery*, *5*(2), 136–146. https://doi.org/10.1002/rcs.238

Garg, A., Sen, S., Kapadia, R., Jen, Y., McKinley, S., Miller, L., & Goldberg, K. (2016). Tumor localization using automated palpation with Gaussian Process Adaptive Sampling. *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, 194–200. https://doi.org/10.1109/COASE.2016.7743380

Gerboni, G., Brancadoro, M., Tortora, G., Diodato, A., Cianchetti, M., & Menciassi, A. (2016). *A novel linear elastic actuator for minimally invasive surgery: Development of a surgical gripper*. http://iopscience.iop.org/article/10.1088/0964-1726/25/10/105025/pdf

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 249–256. http://proceedings.mlr.press/v9/glorot10a.html

Goldman, R. (1990). MATRICES AND TRANSFORMATIONS. In A. S. Glassner (Ed.), *Graphics Gems* (pp. 472–475). Morgan Kaufmann. https://doi.org/10.1016/B978-0-08-050753-8.50110-8

Granick, M., Boykin, J., Gamelli, R., Schultz, G., & Tenenhaus, M. (2006). Toward a common language: Surgical wound bed preparation and debridement. *Wound Repair and Regeneration: Official Publication of the Wound Healing Society [and] the European Tissue Repair Society*, *14 Suppl 1*, S1-10. https://doi.org/10.1111/j.1743-6109.2005.00096.x

Grassmann, R., & Burgner-Kahrs, J. (2019). On the Merits of Joint Space and Orientation Representations in Learning the Forward Kinematics in SE(3). *Robotics: Science and Systems*.

Graves, A. (2014). Generating Sequences With Recurrent Neural Networks. *ArXiv:1308.0850 [Cs]*. http://arxiv.org/abs/1308.0850

Graves, A., Mohamed, A., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 6645–6649. https://doi.org/10.1109/ICASSP.2013.6638947

Gribovskaya, E., & Billard, A. (2008). Combining Dynamical Systems control and programming by demonstration for teaching discrete bimanual coordination tasks to a humanoid robot. *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 33–40.

Gui, T., Zhang, Q., Zhao, L., Lin, Y., Peng, M., Gong, J., & Huang, X. (2019). Long Short-Term Memory with Dynamic Skip Connections. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 6481–6488. https://doi.org/10.1609/aaai.v33i01.33016481

Guo, Z., Dong, Z., Lee, K.-H., Cheung, C. L., Fu, H.-C., Ho, J. D. L., He, H., Poon, W.-S., Chan, D. T.-M., & Kwok, K.-W. (2018). Compact Design of a Hydraulic Driving Robot for Intraoperative MRI-Guided Bilateral Stereotactic Neurosurgery. *IEEE Robotics and Automation Letters*, *3*(3), 2515–2522. https://doi.org/10.1109/LRA.2018.2814637

Haarnoja, T., Pong, V., Zhou, A., Dalal, M., Abbeel, P., & Levine, S. (2018). Composable Deep Reinforcement Learning for Robotic Manipulation. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 6244–6251. https://doi.org/10.1109/ICRA.2018.8460756

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hofstad, E. F., Våpenstad, C., Chmarra, M. K., Langø, T., Kuhry, E., & Mårvik, R. (2013). A study of psychomotor skills in minimally invasive surgery: What differentiates expert and nonexpert performance. *Surgical Endoscopy*, *27*(3), 854–863. https://doi.org/10.1007/s00464-012-2524-9

Hoi, S. C. H., Sahoo, D., Lu, J., & Zhao, P. (2018). Online Learning: A Comprehensive Survey. *ArXiv:1802.02871 [Cs]*. http://arxiv.org/abs/1802.02871

Hu, D., Gong, Y., Hannaford, B., & Seibel, E. J. (2015). Semi-autonomous Simulated Brain Tumor Ablation with RavenII Surgical Robot using Behavior Tree. *IEEE International Conference on Robotics and Automation : ICRA : [Proceedings] IEEE International Conference on Robotics and Automation*, *2015*, 3868–3875. https://doi.org/10.1109/ICRA.2015.7139738

Hussein, A., Gaber, M. M., Elyan, E., & Jayne, C. (2017). Imitation Learning: A Survey of Learning Methods. *ACM Computing Surveys*, *50*(2), 1–35. https://doi.org/10.1145/3054912

Hwang, M., Seita, D., Thananjeyan, B., Ichnowski, J., Paradis, S., Fer, D., Low, T., & Goldberg, K. (2020). Applying Depth-Sensing to Automated Surgical Manipulation with a da Vinci Robot. *ArXiv:2002.06302 [Cs]*. http://arxiv.org/abs/2002.06302

Inoue, M., Yamashita, T., & Nishida, T. (2019). Robot Path Planning by LSTM Network Under Changing Environment. In S. K. Bhatia, S. Tiwari, K. K. Mishra, & M. C. Trivedi (Eds.), *Advances in Computer Communication and Computational Sciences* (pp. 317–329). Springer. https://doi.org/10.1007/978-981-13-0341-8_29

James, S., Davison, A. J., & Johns, E. (2017a). Transferring End-to-End Visuomotor Control from Simulation to Real World for a Multi-Stage Task. *ArXiv:1707.02267 [Cs]*. http://arxiv.org/abs/1707.02267

James, S., Davison, A. J., & Johns, E. (2017b). Transferring End-to-End Visuomotor Control from Simulation to Real World for a Multi-Stage Task. *Conference on Robot Learning*, 334–343. http://proceedings.mlr.press/v78/james17a.html

James, S., Freese, M., & Davison, A. J. (2019). PyRep: Bringing V-REP to Deep Robot Learning. *ArXiv:1906.11176 [Cs]*. http://arxiv.org/abs/1906.11176

Kassahun, Y., Yu, B., Tibebu, A. T., Stoyanov, D., Giannarou, S., Metzen, J. H., & Vander Poorten, E. (2016). Surgical robotics beyond enhanced dexterity instrumentation: A survey of machine learning techniques and their role in intelligent and autonomous surgical actions. *International Journal of Computer Assisted Radiology and Surgery*, *11*(4), 553–568.

Kazanzides, P., Chen, Z., Deguet, A., Fischer, G. S., Taylor, R. H., & DiMaio, S. P. (2014). An open-source research kit for the da Vinci (r); Surgical System. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 6434–6439. https://doi.org/10.1109/ICRA.2014.6907809

Kehoe, B., Kahn, G., Mahler, J., Kim, J., Lee, A., Lee, A., Nakagawa, K., Patil, S., Boyd, W. D., Abbeel, P., & Goldberg, K. (2014). Autonomous multilateral debridement with the Raven surgical robot. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 1432–1439. https://doi.org/10.1109/ICRA.2014.6907040

Kehoe, Ben, Kahn, G., Mahler, J., Kim, J., Lee, A., Lee, A., Nakagawa, K., Patil, S., Boyd, W. D., Abbeel, P., & Goldberg, K. (2014). Autonomous multilateral debridement with the Raven surgical robot. *Proceedings - IEEE International Conference on Robotics and Automation*, 1432–1439. https://doi.org/10.1109/ICRA.2014.6907040

keras.io. (2020). *Keras: The Python deep learning API*. https://keras.io/

Khan, N., Abboudi, H., Khan, M. S., Dasgupta, P., & Ahmed, K. (2014). Measuring the surgical 'learning curve': Methods, variables and competency. *BJU International*, *113*(3), 504–508. https://doi.org/10.1111/bju.12197

Kim, B., Farahmand, A., Pineau, J., & Precup, D. (2013). Learning from Limited Demonstrations. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 26* (pp. 2859–2867). Curran Associates, Inc. http://papers.nips.cc/paper/4918-learning-from-limited-demonstrations.pdf

Kim, J. W., He, C., Urias, M., Gehlbach, P., Hager, G. D., Iordachita, I., & Kobilarov, M. (2019). Autonomously Navigating a Surgical Tool Inside the Eye by Learning from Demonstration. *ICRA 2019*. https://par.nsf.gov/biblio/10136845-autonomously-navigating-surgical-tool-inside-eye-learning-from-demonstration

Kim, Y., Cheng, S. S., Diakite, M., Gullapalli, R. P., Simard, J. M., & Desai, J. P. (2017). Toward the Development of a Flexible Mesoscale MRI-Compatible Neurosurgical Continuum Robot. *IEEE Transactions on Robotics*, *33*(6), 1386–1397. https://doi.org/10.1109/TRO.2017.2719035

Kober, J., & Peters, J. (2009). Learning motor primitives for robotics. *2009 IEEE International Conference on Robotics and Automation*, 2112–2118. https://doi.org/10.1109/ROBOT.2009.5152577

Krishnan, S., Garg, A., Patil, S., Lea, C., Hager, G. D., Abbeel, P., & Goldberg, K. (2015). *Unsupervised Surgical Task Segmentation with Milestone Learning*.

Kroemer, O., Niekum, S., & Konidaris, G. (2019). A Review of Robot Learning for Manipulation: Challenges, Representations, and Algorithms. *ArXiv:1907.03146 [Cs]*. http://arxiv.org/abs/1907.03146

Kuefler, A., Morton, J., Wheeler, T., & Kochenderfer, M. (2017). Imitating driver behavior with generative adversarial networks. *2017 IEEE Intelligent Vehicles Symposium (IV)*, 204–211. https://doi.org/10.1109/IVS.2017.7995721

Kundhal, P. S., & Grantcharov, T. P. (2008). Psychomotor performance measured in a virtual environment correlates with technical skills in the operating room. *Surgical Endoscopy*, *23*(3), 645. https://doi.org/10.1007/s00464-008-0043-5

Lepetit, V., & Fua, P. (2005). *Monocular Model-based 3D Tracking of Rigid Objects*. Now Publishers Inc.

Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, *17*(1), 1334–1373.

Li, Y., Hannaford, B., & Rosen, J. (2019). The Raven Open Surgical Robotic Platforms: A Review and Prospect. *Acta Polytechnica Hungarica*, *16*(8).

Liang, J., Mahler, J., Laskey, M., Li, P., & Goldberg, K. (2017). Using dVRK Teleoperation to Facilitate Deep Learning of Automation Tasks for an Industrial Robot. *IEEE CASE*. IEEE International Conference on Automation Science and Engineering (CASE), Xi'an, China.

Liao, H.-J., Dong, C., Kong, F.-J., Zhang, Z.-P., Huang, P., & Chang, S. (2014). The CUSUM Analysis of the Learning Curve for Endoscopic Thyroidectomy by the Breast Approach. *Surgical Innovation*, *21*(2), 221–228. https://doi.org/10.1177/1553350613500722

Lin, H. C., Shafran, I., Yuh, D., & Hager, G. D. (2006). Towards automatic skill evaluation: Detection and segmentation of robot-assisted surgical motions. *Computer Aided Surgery*, *11*(5), 220–230. https://doi.org/10.3109/10929080600989189

Liu, J., Yuan, T., & Zhang, C. (2011). Three cases using platelet-rich plasma to cure chronic soft tissue lesions. *Transfusion and Apheresis Science*, *45*(2), 151–155. https://doi.org/10.1016/j.transci.2011.07.011

Liu, W., & Wang, T. (2012). Online active multi-field learning for efficient email spam filtering. *Knowledge and Information Systems*, *33*(1), 117–136. https://doi.org/10.1007/s10115-011-0461-x

Liu, Yuejiang, Xu, A., & Chen, Z. (2018). Map-based Deep Imitation Learning for Obstacle Avoidance. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 8644–8649. https://doi.org/10.1109/IROS.2018.8593683

Liu, Yueyue, Li, Z., Liu, H., & Kan, Z. (2020). Skill transfer learning for autonomous robots and human–robot cooperation: A survey. *Robotics and Autonomous Systems*, *128*, 103515. https://doi.org/10.1016/j.robot.2020.103515

Liu, YuXuan, Gupta, A., Abbeel, P., & Levine, S. (2018). Imitation from Observation: Learning to Imitate Behaviors from Raw Video via Context Translation. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1118–1125. https://doi.org/10.1109/ICRA.2018.8462901

Loquercio, A., Maqueda, A. I., del-Blanco, C. R., & Scaramuzza, D. (2018). DroNet: Learning to Fly by Driving. *IEEE Robotics and Automation Letters*, *3*(2), 1088–1095. https://doi.org/10.1109/LRA.2018.2795643

Madapana, N., Rahman, M. M., Sanchez-Tamayo, N., Balakuntala, M. V., Gonzalez, G., Bindu, J. P., Venkatesh, L. N. V., Zhang, X., Noguera, J. B., Low, T., Voyles, R., Xue, Y., & Wachs, J. (2019). DESK: A Robotic Activity Dataset for Dexterous Surgical Skills Transfer to Medical Robots. *ArXiv:1903.00959 [Cs]*. http://arxiv.org/abs/1903.00959

Marescaux, J., Leroy, J., Rubino, F., Smith, M., Vix, M., Simone, M., & Mutter, D. (2002). Transcontinental Robot-Assisted Remote Telesurgery: Feasibility and Potential Applications. *Annals of Surgery*, *235*(4), 487–492.

Marinho, M. M., Harada, K., Morita, A., & Mitsuishi, M. (2020). SmartArm: Integration and validation of a versatile surgical robotic system for constrained workspaces. *The International Journal of Medical Robotics + Computer Assisted Surgery: MRCAS*, *16*(2), e2053. https://doi.org/10.1002/rcs.2053

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, … Xiaoqiang Zheng. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. https://www.tensorflow.org/

Mayer, H., Gomez, F., Wierstra, D., Nagy, I., Knoll, A., & Schmidhuber, J. (2006). A System for Robotic Heart Surgery that Learns to Tie Knots Using Recurrent Neural Networks. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 543–548. https://doi.org/10.1109/IROS.2006.282190

McKinley, S., Garg, A., Sen, S., Gealy, D. V., McKinley, J. P., Jen, Y., Guo, M., Boyd, D., & Goldberg, K. (2016). An interchangeable surgical instrument system with application to supervised automation of multilateral tumor resection. *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, 821–826. https://doi.org/10.1109/COASE.2016.7743487

Miller, K., & Curet, M. (2019). Intuitive Surgical: An Overview. In S. Tsuda & O. Y. Kudsi (Eds.), *Robotic-Assisted Minimally Invasive Surgery: A Comprehensive Textbook* (pp. 3–11). Springer International Publishing. https://doi.org/10.1007/978-3-319-96866-7_1

Mintenbeck, J., Siegfarth, M., Estana, R., & Worn, H. (2014). Flexible instrument for minimally invasive robotic surgery using rapid prototyping technology for fabrication. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, 1085–1090. https://doi.org/10.1109/AIM.2014.6878225

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

Mönnich, H., Wörn, H., & Stein, D. (2012). OP sense—A robotic research platform for telemanipulated and automatic computer assisted surgery. *2012 12th IEEE International Workshop on Advanced Motion Control (AMC)*, 1–6. https://doi.org/10.1109/AMC.2012.6197017

Mott, K. (2019). Estimating Forces of Robotic Pouring Using a LSTM RNN. *ArXiv:1904.09980 [Cs]*. http://arxiv.org/abs/1904.09980

Moustris, G. P., Hiridis, S. C., Deliparaschos, K. M., & Konstantinidis, K. M. (2011). Evolution of autonomous and semi-autonomous robotic surgical systems: A review of the literature. *The International Journal of Medical Robotics and Computer Assisted Surgery*, *7*(4), 375–392. https://doi.org/10.1002/rcs.408

Mukherjee, U. K., & Sinha, K. K. (2020). Robot-assisted surgical care delivery at a hospital: Policies for maximizing clinical outcome benefits and minimizing costs. *Journal of Operations Management*, *66*(1–2), 227–256. https://doi.org/10.1002/joom.1058

Murali, A., Sen, S., Kehoe, B., Garg, A., McFarland, S., Patil, S., Boyd, W., Lim, S., Abbeel, P., & Goldberg, K. (2015). Learning by observation for surgical subtasks: Multilateral cutting of 3D viscoelastic and 2D Orthotropic Tissue Phantoms. *Proceedings - IEEE ICRA*, 1202–1209.

Murray, R. M., Li, Z., Sastry, S. S., & Sastry, S. S. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press.

Nagy, T. D., & Haidegger, T. (2019). A DVRK-based Framework for Surgical Subtask Automation. *ACTA POLYTECHNICA HUNGARICA*, 61–78.

Nair, A., McGrew, B., Andrychowicz, M., Zaremba, W., & Abbeel, P. (2018). Overcoming Exploration in Reinforcement Learning with Demonstrations. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 6292–6299. https://doi.org/10.1109/ICRA.2018.8463162

Nguyen, N. D., Nguyen, T., Nahavandi, S., Bhatti, A., & Guest, G. (2019). Manipulating Soft Tissues by Deep Reinforcement Learning for Autonomous Robotic Surgery. *ArXiv:1902.05183 [Cs]*. http://arxiv.org/abs/1902.05183

Nichols, K. A., Murali, A., Sen, S., Goldberg, K., & Okamura, A. M. (2015). Models of human-centered automation in a debridement task. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5784–5789. https://doi.org/10.1109/IROS.2015.7354198

Niehorster, D. C., Li, L., & Lappe, M. (2017). The Accuracy and Precision of Position and Orientation Tracking in the HTC Vive Virtual Reality System for Scientific Research. *I-Perception*, *8*(3), 2041669517708205. https://doi.org/10.1177/2041669517708205

Oropesa, I., Chmarra, M. K., Sánchez-González, P., Lamata, P., Rodrigues, S. P., Enciso, S., Sánchez-Margallo, F. M., Jansen, F.-W., Dankelman, J., & Gómez, E. J. (2013). Relevance of Motion-Related Assessment Metrics in Laparoscopic Surgery. *Surgical Innovation*, *20*(3), 299–312. https://doi.org/10.1177/1553350612459808

Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., & Peters, J. (2018). An Algorithmic Perspective on Imitation Learning. *Foundations and Trends in Robotics*, *7*(1–2), 1–179. https://doi.org/10.1561/2300000053

Osa, T., Sugita, N., & Mitsuishi, M. (2018). Online Trajectory Planning and Force Control for Automation of Surgical Tasks. *IEEE Transactions on Automation Science and Engineering*, *15*(2), 675–691. https://doi.org/10.1109/TASE.2017.2676018

Padoy, N., & Hager, G. D. (2011). Human-Machine Collaborative surgery using learned models. *2011 IEEE International Conference on Robotics and Automation*, 5285–5292. https://doi.org/10.1109/ICRA.2011.5980250

Pairet, È., Ardón, P., Broz, F., Mistry, M., & Petillot, Y. (2019). Learning and Generalisation of Primitives Skills Towards Robust Dual-arm Manipulation. *ArXiv:1904.01568 [Cs]*. http://arxiv.org/abs/1904.01568

Park, E. J., Kim, C. W., Cho, M. S., Baik, S. H., Kim, D. W., Min, B. S., Lee, K. Y., & Kim, N. K. (2014). Multidimensional analyses of the learning curve of robotic low anterior resection for rectal cancer: 3-phase learning process comparison. *Surgical Endoscopy*, *28*(10), 2821–2831. https://doi.org/10.1007/s00464-014-3569-8

Pavlichenko, D., Rodriguez, D., Schwarz, M., Lenz, C., Selvam Periyasamy, A., & Behnke, S. (2018). Autonomous Dual-Arm Manipulation of Familiar Objects. *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, 1–9. https://doi.org/10.1109/HUMANOIDS.2018.8624922

Peng, W., Xing, Y., Liu, R., Li, J., & Zhang, Z. (2019). An automatic skill evaluation framework for robotic surgery training. *The International Journal of Medical Robotics and Computer Assisted Surgery*, *15*(1), e1964. https://doi.org/10.1002/rcs.1964

Pitonakova, L., Giuliani, M., Pipe, A., & Winfield, A. (2018). Feature and Performance Comparison of the V-REP, Gazebo and ARGoS Robot Simulators. In M. Giuliani, T. Assaf, & M. E. Giannaccini (Eds.), *Towards Autonomous Robotic Systems* (pp. 357–368). Springer International Publishing. https://doi.org/10.1007/978-3-319-96728-8_30

Prusa, J. (2016). *3D Printing handbook. User manual for 3D printers: Original Prusa i3 MK2 kit 1.75 mm*.

Pundak, G., & Sainath, T. (2017). Highway-LSTM and Recurrent Highway Networks for Speech Recognition. *Proc. Interspeech 2017*.

Rahmatizadeh, Abolghasemi, P., Behal, A., & Bölöni, L. (2016a). Learning real manipulation tasks from virtual demonstrations using LSTM. *ArXiv Preprint ArXiv:1603.03833*.

Rahmatizadeh, R., Abolghasemi, P., Bölöni, L., & Levine, S. (2018). Vision-Based Multi-Task Manipulation for Inexpensive Robots Using End-to-End Learning from Demonstration. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 3758–3765. https://doi.org/10.1109/ICRA.2018.8461076

Rahmatizadeh, Rouhollah, Abolghasemi, P., Behal, A., & Bölöni, L. (2016b). Learning real manipulation tasks from virtual demonstrations using LSTM. *ArXiv Preprint ArXiv:1603.03833*.

Ramasubramanian, K., & Singh, A. (2019). Deep Learning Using Keras and TensorFlow. In K. Ramasubramanian & A. Singh (Eds.), *Machine Learning Using R: With Time Series and Industry-Based Use Cases in R* (pp. 667–688). Apress. https://doi.org/10.1007/978-1-4842-4215-5_11

Rateni, G., Cianchetti, M., Ciuti, G., Menciassi, A., & Laschi, C. (2015). Design and development of a soft robotic gripper for manipulation in minimally invasive surgery: A proof of concept. *Meccanica*, *50*(11), 2855–2863. https://doi.org/10.1007/s11012-015-0261-6

Ravichandar, H., Polydoros, A. S., Chernova, S., & Billard, A. (2020). Recent Advances in Robot Learning from Demonstration. *Annual Review of Control, Robotics, and Autonomous Systems*, *3*(1), 297–330. https://doi.org/10.1146/annurev-control-100819-063206

Reiley, C. E., & Hager, G. D. (2009). Task versus Subtask Surgical Skill Evaluation of Robotic Minimally Invasive Surgery. In G.-Z. Yang, D. Hawkes, D. Rueckert, A. Noble, & C. Taylor (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009* (pp. 435–442). Springer Berlin Heidelberg.

Reiley, C. E., Lin, H. C., Yuh, D. D., & Hager, G. D. (2011). Review of methods for objective surgical skill evaluation. *Surgical Endoscopy*, *25*(2), 356–366. https://doi.org/10.1007/s00464-010-1190-z

Reiley, C. E., Plaku, E., & Hager, G. D. (2010). Motion generation of robotic surgical tasks: Learning from expert demonstrations. *Conference Proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, *2010*, 967–970. https://doi.org/10.1109/IEMBS.2010.5627594

Richter, F., Orosco, R. K., & Yip, M. C. (2019). Open-Sourced Reinforcement Learning Environments for Surgical Robotics. *ArXiv:1903.02090 [Cs]*. http://arxiv.org/abs/1903.02090

Ritter, E. M., & Scott, D. J. (2007). Design of a Proficiency-Based Skills Training Curriculum for the Fundamentals of Laparoscopic Surgery. *Surgical Innovation*, *14*(2), 107–112. https://doi.org/10.1177/1553350607302329

Rivard, J. D., Vergis, A. S., Unger, B. J., Hardy, K. M., Andrew, C. G., Gillman, L. M., & Park, J. (2014). Construct validity of individual and summary performance metrics associated with a computer-based laparoscopic simulator. *Surgical Endoscopy*, *28*(6), 1921–1928. https://doi.org/10.1007/s00464-013-3414-5

Robotics, A. (2014). Technical reference manual: RAPID instructions, functions and data types. *ABB Robotics, Väster\aas, Sweden*.

Rosen, J., & Ma, J. (2015). Autonomous Operation in Surgical Robotics. *Mechanical Engineering*, *137*(09), S15–S18. https://doi.org/10.1115/1.2015-Sep-9

Sak, H., Senior, A. W., & Beaufays, F. (2014). *Long short-term memory recurrent neural network architectures for large scale acoustic modeling*.

Saleh, K., Attia, M., Hossny, M., Hanoun, S., Salaken, S., & Nahavandi, S. (2018). Local Motion Planning for Ground Mobile Robots via Deep Imitation Learning. *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 4077–4082. https://doi.org/10.1109/SMC.2018.00691

Sanchez-Tamayo, N., & Wachs, J. P. (2018). Collaborative Robots in Surgical Research: A Low-Cost Adaptation. *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 231–232. https://doi.org/10.1145/3173386.3176978

Schulman, J., Gupta, A., Venkatesan, S., Tayson-Frederick, M., & Abbeel, P. (2013). A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4111–4117. https://doi.org/10.1109/IROS.2013.6696945

Seita, D., Krishnan, S., Fox, R., McKinley, S., Canny, J., & Goldberg, K. (2018). Fast and Reliable Autonomous Surgical Debridement with Cable-Driven Robots Using a Two-Phase Calibration Procedure. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 6651–6658. https://doi.org/10.1109/ICRA.2018.8460583

Sen, S., Garg, A., Gealy, D. V., McKinley, S., Jen, Y., & Goldberg, K. (2016). Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 4178–4185. https://doi.org/10.1109/ICRA.2016.7487611

Shademan, A., Decker, R. S., Opfermann, J. D., Leonard, S., Krieger, A., & Kim, P. C. W. (2016). Supervised autonomous robotic soft tissue surgery. *Science Translational Medicine*, *8*(337), 337ra64-337ra64. https://doi.org/10.1126/scitranslmed.aad9398

Sheng, J., Wang, X., Dickfeld, T.-M. L., & Desai, J. P. (2018). Towards the Development of a Steerable and MRI-Compatible Cardiac Catheter for Atrial Fibrillation Treatment. *IEEE Robotics and Automation Letters*, *3*(4), 4038–4045. https://doi.org/10.1109/LRA.2018.2861011

Shin, C., Ferguson, P. W., Pedram, S. A., Ma, J., Dutson, E. P., & Rosen, J. (2019). Autonomous Tissue Manipulation via Surgical Robot Using Learning Based Model Predictive Control. *2019 International Conference on Robotics and Automation (ICRA)*, 3875–3881. https://doi.org/10.1109/ICRA.2019.8794159

Smith, C., Karayiannidis, Y., Nalpantidis, L., Gratal, X., Qi, P., Dimarogonas, D. V., & Kragic, D. (2012). Dual arm manipulation—A survey. *Robotics and Autonomous Systems*, *60*(10), 1340–1353. https://doi.org/10.1016/j.robot.2012.07.005

Solà, J. (2017). Quaternion kinematics for the error-state Kalman filter. *ArXiv:1711.02508 [Cs]*. http://arxiv.org/abs/1711.02508

Steed, D. L. (2004). Debridement. *The American Journal of Surgery*, *187*(5, Supplement 1), S71–S74. https://doi.org/10.1016/S0002-9610(03)00307-6

Sun, W., Venkatraman, A., Gordon, G. J., Boots, B., & Bagnell, J. A. (2017). Deeply AggreVaTeD: Differentiable imitation learning for sequential prediction. *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, 3309–3318.

Taghbalout, M., Antoine, J. F., & Abba, G. (2019). Experimental Dynamic Identification of a YuMi Collaborative Robot. *IFAC-PapersOnLine*, *52*(13), 1168–1173. https://doi.org/10.1016/j.ifacol.2019.11.354

Tamei, T., Matsubara, T., Rai, A., & Shibata, T. (2011). Reinforcement learning of clothing assistance with a dual-arm robot. *2011 11th IEEE-RAS International Conference on Humanoid Robots*, 733–738. https://doi.org/10.1109/Humanoids.2011.6100915

Taylor, M. E., Suay, H. B., & Chernova, S. (2011). Integrating reinforcement learning with human demonstrations of varying ability. *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, 617–624.

Thananjeyan, B., Garg, A., Krishnan, S., Chen, C., Miller, L., & Goldberg, K. (2017). Multilateral surgical pattern cutting in 2D orthotropic gauze with deep reinforcement learning policies for tensioning. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2371–2378. https://doi.org/10.1109/ICRA.2017.7989275

Tobergte, A. (2010). MiroSurge ;Advanced User Interaction Modalities in Minimally Invasive Robotic Surgery. *Presence*, *19*(5), 400–414. https://doi.org/10.1162/pres_a_00022

Varadarajan, B., Reiley, C., Lin, H., Khudanpur, S., & Hager, G. (2009). Data-Derived Models for Segmentation with Application to Surgical Assessment and Training. In G.-Z. Yang, D. Hawkes, D. Rueckert, A. Noble, & C. Taylor (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009* (pp. 426–434). Springer. https://doi.org/10.1007/978-3-642-04268-3_53

Vecerik, M., Hester, T., Scholz, J., Wang, F., Pietquin, O., Piot, B., Heess, N., Rothörl, T., Lampe, T., & Riedmiller, M. (2018). Leveraging Demonstrations for Deep Reinforcement Learning on Robotics Problems with Sparse Rewards. *ArXiv:1707.08817 [Cs]*. http://arxiv.org/abs/1707.08817

Vedula, S. S., Malpani, A., Ahmidi, N., Khudanpur, S., Hager, G., & Chen, C. C. G. (2016). Task-Level vs. Segment-Level Quantitative Metrics for Surgical Skill Assessment. *Journal of Surgical Education*, *73*(3), 482–489. https://doi.org/10.1016/j.jsurg.2015.11.009

Wang, T., Yang, C., Kirchner, F., Du, P., Sun, F., & Fang, B. (2019). Multimodal grasp data set: A novel visual–tactile data set for robotic manipulation. *International Journal of Advanced Robotic Systems*, *16*(1), 1729881418821571. https://doi.org/10.1177/1729881418821571

Wang, Y., Ye, X., Yang, Y., & Zhang, W. (2017). Collision-free trajectory planning in human-robot interaction through hand movement prediction from vision. *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 305–310. https://doi.org/10.1109/HUMANOIDS.2017.8246890

Watanabe, K., Kanno, T., Ito, K., & Kawashima, K. (2016). Human-integrated automation of suturing task with one-master two-slave system for laparoscopic surgery. *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 1180–1185. https://doi.org/10.1109/AIM.2016.7576930

Weisstein, E. W. (2020). *Affine Transformation* [Text]. Wolfram Research, Inc. https://mathworld.wolfram.com/AffineTransformation.html

Xingguang, Z., P. B., J., & Madapana, N. (2019). *Forward_Project/AnnotationToolPlus* [Project repository]. GitHub. https://github.com/nmadapan/Forward_Project

Yamaguchi, T., Kinugasa, Y., Shiomi, A., Sato, S., Yamakawa, Y., Kagawa, H., Tomioka, H., & Mori, K. (2015). Learning curve for robotic-assisted surgery for rectal cancer: Use of the cumulative sum method. *Surgical Endoscopy*, *29*(7), 1679–1685. https://doi.org/10.1007/s00464-014-3855-5

Yip, M., & Das, N. (2017). Robot Autonomy for Surgery. *ArXiv:1707.03080 [Cs]*. http://arxiv.org/abs/1707.03080

Yoshida, M., Kakushima, N., Mori, K., Igarashi, K., Kawata, N., Tanaka, M., Takizawa, K., Ito, S., Imai, K., Hotta, K., Ishiwatari, H., Matsubayashi, H., & Ono, H. (2017). Learning curve and clinical outcome of gastric endoscopic submucosal dissection performed by trainee operators. *Surgical Endoscopy*, *31*(9), 3614–3622. https://doi.org/10.1007/s00464-016-5393-9

Yu, S., Chen, X., Yang, L., Wu, D., Bennis, M., & Zhang, J. (2020). Intelligent Edge: Leveraging Deep Imitation Learning for Mobile Edge Computation Offloading. *IEEE Wireless Communications*, *27*(1), 92–99. https://doi.org/10.1109/MWC.001.1900232

Yuba, H., Arnold, S., & Yamazaki, K. (2017). Unfolding of a rectangular cloth from unarranged starting shapes by a Dual-Armed robot with a mechanism for managing recognition error and uncertainty. *Advanced Robotics*, *31*(10), 544–556. https://doi.org/10.1080/01691864.2017.1285722

Zhang, D., Chen, J., Li, W., Bautista Salinas, D., & Yang, G.-Z. (2020). A microsurgical robot research platform for robot-assisted microsurgery research and training. *International Journal of Computer Assisted Radiology and Surgery*, *15*(1), 15–25. https://doi.org/10.1007/s11548-019-02074-1

Zhang, T., McCarthy, Z., Jow, O., Lee, D., Chen, X., Goldberg, K., & Abbeel, P. (2018). Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 5628–5635. https://doi.org/10.1109/ICRA.2018.8461249

Zhang, X., Li, W., Chiu, P. W. Y., & Li, Z. (2020). A Novel Flexible Robotic Endoscope With Constrained Tendon-Driven Continuum Mechanism. *IEEE Robotics and Automation Letters*, *5*(2), 1366–1372. https://doi.org/10.1109/LRA.2020.2967737

Zhou, T., Cabrera, M. E., Wachs, J. P., Low, T., & Sundaram, C. (2016). A Comparative Study for Telerobotic Surgery Using Free Hand Gestures. *J. Hum.-Robot Interact.*, *5*(2), 1–28. https://doi.org/10.5898/JHRI.5.2.Zhou

Zhou, Y., Barnes, C., Lu, J., Yang, J., & Li, H. (2019). *On the Continuity of Rotation Representations in Neural Networks*. 5745–5753. http://openaccess.thecvf.com/content_CVPR_2019/html/Zhou_On_the_Continuity_of_Rotation_Representations_in_Neural_Networks_CVPR_2019_paper.html

Zhu, Y., Wang, Z., Merel, J., Rusu, A., Erez, T., Cabi, S., Tunyasuvunakool, S., Kramár, J., Hadsell, R., Freitas, N. de, & Heess, N. (2018). *Reinforcement and Imitation Learning for Diverse Visuomotor Skills*. *14*. http://www.roboticsproceedings.org/rss14/p09.html

VITA

Natalia Sanchez-Tamayo
School of Industrial Engineering, Purdue University

Education

B.S., Mechanical Engineering, 2016, Universidad de los Andes Bogotá, Colombia.

B.S., Civil Engineering, 2017, Universidad de los Andes Bogotá, Colombia.

M.S., Industrial Engineering, 2020. Purdue University, West Lafayette IN, USA.

Research Interests

Human-Robot Interaction, Haptics, Machine Learning, Robot Learning, Autonomous Systems.

PUBLICATIONS

Edgar Rojas-Muñoz, Maria Eugenia Cabrera, Chengyuan Lin, **Natalia Sanchez-Tamayo**, Dan Andersen, Voicu Popescu, Kathryn Anderson, MSGH, Ben Zarzaur, Brian Mullis, Juan P Wachs. (February 2020). Telementoring in Leg Fasciotomies via Mixed-Reality: Clinical Evaluation of the STAR Platform. *Military Medicine*, Volume 185, Issue Supplement_1, January-February 2020, Pages 513–520,

Edgar Rojas-Muñoz, Maria Eugenia Cabrera, **Natalia Sanchez-Tamayo,** Dan Andersen, Voicu Popescu, Juan Antonio Barragan Noguera, Ben Zarzaur, Pat Murphy, Kathryn Anderson, Thomas Douglas, Clare Griffis, and Juan P. Wachs. (March 2020). How About the Mentor? Effective Workspace Visualization in AR Telementoring. 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)

Edgar Rojas-Muñoz, Chengyuan Lin, **Natalia Sanchez-Tamayo**, Maria Eugenia Cabrera, Daniel Andersen, Voicu Popescu, Juan Antonio Barragan, Ben Zarzaur, Patrick Murphy, Kathryn Anderson, Thomas Douglas, Clare Griffis, Jessica McKee, Andrew W. Kirkpatrick & Juan P. Wachs. (May 2020). Evaluation of an augmented reality platform for austere surgical telementoring: a randomized controlled crossover study in cricothyroidotomies. *npj Digital Medicine.*

Naveen Madapana, Md Masudur Rahman, **Natalia Sanchez-Tamayo,** Mythra V. Balakuntala, Glebys Gonzalez, Jyothsna Padmakumar Bindu, Vishnunandan Venkatesh, Xingguang Zhang, Juan Barragan, Thomas Low, Richard M. Voyles, Yexiang Xue, Juan P. Wachs. (November 2019) "DESK: A Robotic Activity Dataset for Dexterous Surgical Skills Transfer to Medical Robots". IEEE/RSJ International conference on Intelligent Robots and Systems (IROS).

Md Masudur Rahman, **Natalia Sanchez-Tamayo,** Glebys Gonzalez, Mridul Agarwal, Vaneet Aggarwal, Richard M. Voyles, Yexiang Xue, Juan P. Wachs. (October 2019). "Transferring Dexterous Surgical Skill Knowledge Between Robots for Semi-Autonomous Operation". IEEE International Conference on Robot & Human Interactive Communication (Ro-Man).

Chengyuan Lin, Edgar Rojas-Munoz, Maria E. Cabrera, **Natalia Sanchez-Tamayo**, Daniel Andersen, Voicu Popescu, Juan Antonio Barragan Noguera, Ben Zarzaur, Pat Murphy, Kathryn Anderson, Thomas Douglas, Clare Griffis, and Juan P. Wachs. (March 2019). "Robust High-Level Video Stabilization for Effective AR Telementoring". IEEE Conference on Virtual Reality and 3D User Interfaces (VR).

Natalia Sanchez-Tamayo, & Juan P. Wachs. (March 2018). "Collaborative Robots in Surgical Research: a Low-Cost Adaptation" ACM/IEEE International Conference on Human-Robot Interaction (HRI).

Maria E. Cabrera, **Natalia Sanchez-Tamayo**, Richard Voyles, & Juan P. Wachs (March 2018). "One-Shot Gesture Recognition: One Step Towards Adaptive Learning". IEEE International Conference on Automatic Face & Gesture Recognition (FG).