Imitating Human Regrasping Movements with a Robotic Hand using Tangle Topology

タングルトポロジーを用いたロボットハンドによる
人間の持ち替え動作の模倣

ビナヤウェキン　ポンタリン

Imitating Human Regrasping Movements with a Robotic Hand using Tangle Topology

(タングルトポロジーを用いたロボットハンドによる
人間の持ち替え動作の模倣)

Phongtharin Vinayavekhin

ビナヤウェキン　ポンタリン

Dissertation

Submitted in partial fulfillment of the requirement
for the degree of Doctor of Philosophy
in Information Science and Technology

December 2012

Imitating Human Regrasping Movements with a Robotic Hand using Tangle Topology

(タングルトポロジーを用いたロボットハンドによる
人間の持ち替え動作の模倣)

Phongtharin Vinayavekhin

ビナヤウェキン　ポンタリン

Dissertation Committee:

Kiyoharu AIZAWA　(Chair)
Mitsuru ISHIZUKA
Yoichi HORI
Takashi KUBOTA
Yoichi SATO

Dissertation Supervisor:

Katsushi IKEUCHI

I would like to dedicate this work to my family;
to dad, whose devotion has always provided for us;
to mum, who has made countless sacrifices for us;
and to Yui, who has always been the perfect role model.

ABSTRACT

There have been many developments of anthropomorphic robotic hand in recent years. These hands have become very advanced in term of the hardware. They have also become more and more similar to human hand. Unfortunately, there are only a few evidences that have displayed the capabilities of their usages due to the lack of efficient method to control them. This thesis proposes a methodology to control these robotic hands through human demonstration. The method focuses on a dexterous manipulation of a hand referred to as regrasping. Regrasping is an ability of a hand to change its grasping posture with an object by moving the fingers. The proposed method teaches a robot to imitate these movements. It allows a robot to observe a human perform regrasping movements, recognise, and finally reproduce the regrasping movement.

The proposed method is based on a representation referred to as tangle topology. Tangle topology is a representation that derived from a numerical invariant that describes a relation between two curves called Gauss Linking Integral (GLI). When hand and a manipulated object are considered as strands, it allows regrasping movements to be perceived as a change of tangle relation over time. The tangle relation is described by an attribute called *writhe matrix*. Writhe matrix is categorised into two types, which distinguish a contact relation between the hand and object. Using this topological information, *movement primitives* for recognising regrasping movement and *skill parameters* for mapping each movement primitive to a robotic hand are defined.

Three movement primitives are defined based on changes in type of writhe matrices during regrasping. The regrasping movement is first segmented into smaller segments using parameters that describe writhe matrix. Then by considering the change in type of writhe matrices, each small segment is recognised into a movement primitive. Once all movement primitives are recognised, the original regrasping movement can then be represented as a sequence of movement primitives.

To reproduce a regrasping movement, all movement primitives that represent the movement must be sequentially mapped to a robotic hand. Skill parameters for all movement primitive are observed and extracted from human demonstra-

tion. They are key information for mapping the movement primitives to a robotic hand. Skill parameters are defined differently for each type of movement primitive. A method to refine the observed skill parameters is also given to make them suitable to be used during mapping process. Movement primitives are mapped to a robotic hand in topological space. A method to interpolate writhe matrix from its initial to final state is given. A trajectory of the hand for each movement primitives is generated by following these intermediate writhe matrices.

The proposed methodology is validated by reproducing a regrasping movement in a robotic hand. Human regrasping movements of a pen-like object are considered. A custom-made robotic hand is attached to the Mitsubishi PA-10 robotic arm to maneuver and reproduce the movements. The successful reproduction of the regrasping movement verifies the proposed methodology to be useful and proved that it is feasible to control a robotic hand by imitating human.

In short, this thesis describes a novel method to control a robotic hand by imitating human movement. It focuses on hand movements referred to as regrasping movements. The thesis shows that by observing and representing human movements as a sequence of movement primitives, a robot can duplicate the movement on its hand.

ACKNOWLEDGMENTS

Completing my PhD degree is probably one of the most challenging achievements of the first 28 years of my life. This doctoral dissertation would never have come to fruition without the support of many individuals. I would like to take this opportunity to show my appreciation to all those persons who have given their invaluable support and assistance.

First and foremost, I would like to express my sincere gratitude to my dissertation supervisor, Prof. Katsushi Ikeuchi, for his consideration and supervision throughout my graduate school. His vision on the setting of this study allowed me to work in an unexplored and challenging topic. His understanding in scientific research has taught me what it needs to conduct a research properly. I am also grateful to Dr. Jun Takamatsu, and my dissertation committee, Prof. Kiyoharu Aizawa, Prof. Mitsuru Ishizuka, Prof. Yoichi Hori, Prof. Takashi Kubota, and Prof. Yoichi Sato, for providing valuable feedbacks and advices on my work.

Dr. Shunsuke Kudoh is someone who cannot be missed out on my appreciation list. His vision, advices, and supports allowed me to proceed through the doctoral program and complete this dissertation. His suggestion always help me decides the priority; discussing with him is always helpful and encouraging. I was very fortunate to work with him. I also would like to thank Dr. Koichi Ogawara and Mr. Yoshihiro Sato for providing their great assistances in constructing and repairing Dotchan.

I am very grateful to Björn Rennhak and Kent Fujiwara for their advices and fruitful discussions on both my research works and other topics. I definitely would not be able to endure the last two years of my study without you guys. I also would like to thank Lu Min and Takahiro Okamoto for their technical discussions. Other (past/present) CVL members are also deserved to be added into the list for their helps with my work one way or the other: Dr. Takeshi Oishi, Dr. Bo Zheng, Shangbin Shu, Mawo Kamakura, Osamu Naito, Yoshie Kobayashi, Hiroki Yamanaka, other CVL staffs and secretaries.

I have been fortunate to have great friends in Japan, whose support was invaluable. In particular, I would like to thank my flat mate and neighbors at Avenue Seijou 2, namely Jason, Matt and Mostafa. For the last two years,

you guys have made the place a home. I am also appreciated friendships from Soshigaya Söderling Appreciation Society, that encouraged me to exercise more and also taught me to control my nerves.

Finally, I would like to thank my parents for giving me this amazing life; my sister for her support and understanding. I owe them everything and wish I could show them just how much I love and appreciate them. Without your love, support and encouragement, I would not have come this far. I hope that this work makes you proud.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER

ONE

Introduction

Contents

1.1 Background

It was in 1954 when George C. Devol applied for a patent for *"a more or less general purpose machine that has universal application to a vast diversity of applications where cyclic digital control is desired."*, that human interest in robotic arm-like devices first began. Since then, robots have been widely used in factories around the world to automate a manufacturing process and free human from repetitive, risky and tiresome tasks. Applications in an industry has dominated the usages of robots for (quite) some time, until recently when a trend of service robots has emerged [Garcia *et al.* 2007] that robots have started to move into human environments.

There are many potentials for robots to assist human in everyday settings, e.g. accommodating the elderly, helping in a household or a restaurant, working in hazardous or contaminated environments. These applications have been studied and researched in the laboratories around the world, but only a few accomplished

to become commercially available. Most promising applications need robots to be able to automatically interact and physically alter the world. However, the ability of a robot to perform sophisticated manipulation tasks is only possible in a controlled environment or when controlled by a human [Kemp *et al.* 2007].

There are many ways for a robot to interact with the world. One of the ways is to use a *hand*, which is biologically inspired by human. A hand is an important human body part. Throughout a day, human uses its hands numerous times, either with prehensile or non-prehensile movements [Jones & Lederman 2006], to alter the environment. A robot that possesses a hand with such dexterity as human hand would be able to interact well with its physical surroundings.

Many anthropomorphic robotic hands has been developed in recent years, e.g. the Utah/MIT hand [Jacobsen *et al.* 1984], the Belgrade/USC hand [Bekey *et al.* 1990], the NASA Robonaut hand [Lovchik *et al.* 1999], the DLR hand [Butterfass *et al.* 1998], the Rutgers hand [DeLaurentis *et al.* 2000], the Shadow hand [Reichel 2004]. With the advancement in technologies that used to construct them, these hands have become very complex in term of their hardware functionalities and have also become more and more similar to a human hand. Unfortunately, there are only a few evidences that has displayed the capabilities of their usages in everyday environment, except for those of the teleoperation application. One of the main issues is the lack of efficient approach to control these robot hands.

This thesis focuses an approach to control a robotic hand by observing how human move their hand. This is based on the assumption that if the robotic hands were to build to emulate how human hands are maneuvered and manipulate objects, the best way to control the robot hand is to simply imitate human. Although the imitation approach might seem reasonable, in reality, it is not as trivial and straightforward as it may sound. The most critical question is how to model and represent human hand movement, so that it can be applied to robot hands. Since a general human hand movement is very complicated and highly sophisticated, this thesis will focus on one type of human dexterous manipulative movement called *regrasping* or sometimes referred to a *in-hand manipulation*.

Regrasping is an ability of a hand to change its grasping posture with an object by moving the fingers or, in other words changing the contact location of fingers. Humans usually have more than one grasping posture for one type of tool. A human makes a decision on which type of grasping posture to use based on their intention of how to use the grasped object at that particular moment [Cutkosky 1989]. For example, an artist grasps a pen or a paintbrush differently depending on whether they want to draw or measure the length of the scene, or people also grasp a hammer differently depending on whether they want to strike or pull out a nail, or people grasp mobile phones differently depending on whether they want to press a button or talk on the phone. This makes regrasping an important function for robotic hands, if they were to be used to

interact with tools created for humans.

1.2 Approaches in Regrasping Planning

Giving two grasping configurations, initial and final grasping configurations, regrasping planning is defined as a problem of finding trajectories of hand and its joint angles to move from one configuration to the other. Several approaches have been proposed in the literature to complete a regrasping movement in a robotic hand. They can be divided into three major categories: a.) teleoperation, b.) automatic programming, and c.) learning from human.

1.2.1 Teleoperation Approach

A teleoperation is the most intuitive approach to control a robotic hand. It is usually referred to as a master-slave architecture of a control system, where a human directly controls a slave robot hand to manipulate an object through a master device. There are applications in many areas for this control scheme. For example, a robotic surgical system in medical [Bann *et al.* 2003, Camarillo *et al.* 2004], a maintenance humanoid robot in space [Ambrose *et al.* 2000, Diftler *et al.* 2003], a mobile manipulation robot in hazardous areas [Taylor 1985].

This approach has been studied extensively in the last few decades. With some efforts from human operator, the approach can be used to successfully perform many complex manipulation tasks [Kemp *et al.* 2007]. Relatively recent improvements for this approach can be seen in the following areas: precision and ease of use [Fischer *et al.* 1998, Hu *et al.* 2004], haptic interface and force feedback [Martin *et al.* 2004, Endo *et al.* 2011, Lii *et al.* 2010], Electromyography interface [Farry *et al.* 1996, Bitzer & van der Smagt 2006].

1.2.2 Automatic Programming Approach

Automatic programming is the most common approach in the literature. In this approach, regrasping planning problem is usually divided into two subproblems [Trinkle & Hunter 1991, Michelman 1998]: a problem of searching for a path in a configuration space of a specific representation, and a problem of generating trajectories of hand/fingers to achieve the path. Both problem can be solved either separately or simultaneously.

In this problem, the size of the search space is usually a main concern. By formulating the problem using some specific representation (e.g. contact, or their derivation), the search space becomes smaller and manageable. Various constraints, including collision avoidance, grasp equilibrium etc. are also used to confine the search space. Then, the algorithms are proposed to search a path within the space to connect two grasping configurations together.

If objects are represented by one or a combination of fundamental elements, i.e. vertices, edges, and surface patches, contacts between them can be described by contact formations (CFs) [Desai & Volz 1989]. A grasping configuration is simply a combination of CFs between a hand and an object. Trinkle and Hunter describes initial and final grasping configurations with two CFs, where their corresponding CF-trees are created by sampling an input trajectory of the hand [Trinkle & Hunter 1991]. The proposed method searches for a path to connect the two CFs by branching through CF-trees until the common CF are met and connected. Zhang *et al.* creates a taxonomy of grasping configuration based on all feasible CFs between a hand and an object and referred to it as a grasp transformation graph (GTG) [Zhang *et al.* 1996]. This graph can be used to search for a path to connect between the initial and final grasp configurations.

Han and Trinkle proposed a general framework for dexterous manipulation [Han & Trinkle 1998]. In the framework, a combination of rolling and finger gaiting is used to change from initial to final configuration of point contacts. Rolling is used in general, except when some finger reach its limit, finger gaiting is used instead in order to replace the finger. The proposed framework is applied to manipulate a sphere with three fingertips. Cherif and Gupta allowed finger rolling and sliding to reorient an objects [Cherif & Gupta 1999, Cherif & Gupta 2001]. A dexterous manipulation of four fingers were considered, where only one finger was allowed to move at an instance. The proposed method used A* search to search for a path in a discrete representation of a configuration space, referred to as cell decomposition. Yashima *et al.* proposed a manipulation planner using the Rapidly-Exploring Random Tree (RRT) algorithm [Yashima *et al.* 2003]. At an instance, a choice of motions that a finger can move depends on its contact mode. The proposed planner used this notion to explore and search a configuration space for a regrasping path. Sudsang and Phoka proposed a regrasping planner for four fingered hand [Sudsang & Phoka 2003]. The planner is based on a structure called a switching graph, where each node in the graph represents a grasp that satisfied properties of a concurrent grasp. The proposed method finds the regrasping path by searching in the graph structure. Saut *et al.* proposed a manipulation planner based on a probabilistic roadmap method (PRM) [Saut *et al.* 2006, Saut *et al.* 2007]. The method is based on an ability to search and connect between grasp subspaces, where a grasp subspace is a space of all grasps that can be achieved by a particular set of grasping fingers. Two possible types of path are allowed: transfer path (connecting within the subspace) and regrasping path (connecting between two subspaces). Both object and finger trajectories are given as an output of the planner.

1.2.3 Learning from Human Approach

An approach to learn dexterous manipulation from human is quite new and currently gaining more attention from the community. When referring to this approach in general, it is not only teach a robot to mimic a human hand movement, but also teach it to learn and understand the task that is being imitated [Bicchi 2000]. Unfortunately, to the best of our knowledge, there are only a few researches that are directly related to regrasping planning or dexterous manipulation planning. In order to increase an understanding of the approach, related literatures on grasping planning are also described.

Briefly, planning a grasp for a robotic hand by observing human contains two important components: a.) what to observe and how to represent knowledge from human, b.) how to map the knowledge to a robotic hand. Some frameworks explained below are completed, while some only proposed the former components.

Cutkosky was the first person to suggest that knowledges from human could be useful to plan a grasp for a robotic hand [Cutkosky 1989]. He suggested that apart from a stability, a good grasp must also embrace knowledges about a task requirement, which is already included in a human demonstration. Cutkosky collected the knowledges by observing and interviewing the machinists while they were working and proposed a grasp taxonomy upon it. A rule-based decision system were developed to predict an appropriate grasp to use with a particular object in a specific task environment. Although more details planning on recognition and mapping to a robotic hand was not given, the work has laid a foundation for many other researches.

Kang and Ikeuchi proposed a method to recognise and map a grasp to a robotic hand [Kang & Ikeuchi 1993, Kang & Ikeuchi 1997]. Contacts between a human hand and the object are observed by processing images obtained from TV cameras. Contacts are combined into a representation called contact-web, which a grasp taxonomy is defined upon it. An observed grasp is recognised into one of the grasp in the taxonomy and mapped onto a robotic hand. The mapping is conducted in two levels: the functional level where a concept of virtual finger [Arbib *et al.* 1985] is used, and the physical level where the geometric properties of the object and the hand are considered to fine-tune the grasp. Aleotti and Caselli took a similar approach in planning a grasp from human demonstration [Aleotti & Caselli 2006, Aleotti & Caselli 2007]. However, a demonstration is done in a virtual environment, as it allows more accurate information including contact and joint angles to be collected. Both of which are used to recognise a grasp against a subset of Cutkosky's taxonomy. A mapping to a robotic hand is done using only a recognised grasp type. A posture database of a target robotic hand for every grasp types is captured off-line prior to the mapping. When the human grasp type is identified, a pre-defined posture is fetched and mapped to the robotic hand. Steffen *et al.* also captured joint angles and contact information from a virtual environment [Steffen *et al.* 2007]. However, the information is not

used to recognise a grasp type, but instead used to search for a similar robotics grasp in a database using a structure called a Partial Contact Posture (PCP) as an index. To map the grasp, the fetched grasp is used to initialise a posture of the robotic hand while tactile sensors and finger closing heuristic are used to finalise the grasp.

Ritter *et al.* discovered positions of all fingertips of human hand using images [Nolker & Ritter 2002, Steil *et al.* 2004]. The fingertip positions is transformed into the joint angle of the human hand using a Parameterized Self-organizing Map (PSOM). The joint angles is then mapped to joint angles of the robotic hand by considering the difference of their kinematic constraints. This posture is used to initialise a posture of a robotic hand. The grasp is finalised by moving fingers in the caging fashion while evaluating contact condition on the way. Positions of fingertips can be used to recognise grasp. Ekvall and Kragic proposed a method to recognise a grasp type in Cutkosky's taxonomy using fingertip positions, hand rotation and hand trajectory [Ekvall & Kragic 2005].

Kjellstrom *et al.* took an advantage of vision technology and managed to recognize a type of grasp using a single 2D image [Kjellstrom *et al.* 2008]. The vision-based grasp recognition system can take an input image from any directions, and search in a database to define the types of grasp. Each type of grasp is mapped to robot hand through pre-defined rules. Each rule describes a robot hand posture and roughly a place where to grasp at an object. A similar approach is used by Do *et al.* to plan a grasp on humanoid robot in real time [Do *et al.* 2009].

It can be noticed that most approaches in grasp planning from observation use a grasp taxonomy as a representation to connect between a human and a robot grasp. Similarly, there are also two important components in the dexterous manipulation planning from observation: a.) a representation and recognition phase, b.) a mapping phase. However, same grasp taxonomy might not be a good representation as it is a taxonomy for every kind of objects. For a specific object, the taxonomy could be far too incomplete to represent all hand postures.

Steffen *et al.* [Steffen *et al.* 2008] use a continuous manifold of hand postures to represent a regrasping movement. They construct a manifold by applying Unsupervised Kernel Regression (UKR) and its modification [Steffen *et al.* 2009] on finger joint angle space. The method is applied to turning bottle cap in their example. A demonstration is done using various sizes of bottle cap in a virtual environment and the movement of the same robot hand is automatically reproduced with different cap sizes. They extend their UKR representation framework to closed-loop control and use it to imitate the task of swapping Chinese health balls where balls position are used as feedback [Steffen *et al.* 2010]. The drawback of their framework is that the hands using during demonstration and reproduction must be the same hand.

To overcome the hand structure dependency, the representation is generally

derived from a contact information, either on the hand or on the object. Lam *et al.* combined information observed from human demonstration with motion planning algorithms to recover a sequence of contact points on the object that represents the original movement [Lam *et al.* 1999]. Human demonstrates a regrasping movement using data-glove and motion tracker. In order to recover contact points in every frames, he starts by recovering initial and final contact points based on observed data from data-glove. Once initial and final contact points are found, contact points on the object of all intermediate frames is generated using motion planning while the demonstrated movement is used to limit the search region of the algorithm. Kondo *et al.* attaches a tactile sensor sheet on an object in order to observe changes in contact state during human regrasping movement [Kondo *et al.* 2006, Kondo *et al.* 2008]. They manage to discover a contact state on a human hand at a particular moment by processing a pressure distribution image. A state transition diagram is created based on these contact states. A target robotic hand is preprogrammed with the movement that changes from one contact state to another. Human regrasping movement is recognised and represented as a transition of contact states using Dynamic Programming algorithm. The transition of contact states is sent as a command to the robotic hand to reproduce the movement. Martins *et al.* came up with a similar approach but a tactile sensor is directly attached to a human hand [Martins *et al.* 2010, Faria *et al.* 2012].

It can be seen that for a dexterous manipulation planning from observation, most frameworks only pass a high level knowledge to map to a robotic hand. Other knowledge obtained from the observation has been abstracted out and not used during mapping at all.

1.3 What is This Thesis About?

In a high-level context of learning from human, when a task of everyday object manipulation is taught to the robot, there are many phases recognised and detected. One of the phase that is crucial to the task as a whole is a manipulation phase where human grasps a tool and use it to achieve something. For a particular tool there could be more than one type of feasible grasps. If the hand employs different type of grasps, they are considered to be two different manipulation phases. A regrasping phase connects between two manipulation phases that employ different type of grasps on a particular tool [Kudoh *et al.* 2008]. This thesis focuses solely on developing a method to automatically generate a movement of a robotic hand that will achieve a regrasping phase.

As mentioned earlier in the section, a method is developed based on a Learning from Observation (LFO) paradigm. Initial and final grasps are given as an input from a high-level task planner. A hand movement of human that connects between two grasps is also given as an input. The method then automatically

generates a movement of a robotic hand using knowledges extracted from this human hand movement. The method is based on the fact that many robotic hands are built to resemble human hand; therefore, a movement of a human hand should also be useful for generating a movement for these robotic hands. An advantage of the method is also similar to those that make use of a LFO paradigm. Once the system has been successfully deployed, it would be easy to teach other hand movements to the robot. A human operator can simply demonstrate natural hand movements, and the robot will recognise and imitate them.

It should be noticed that the proposed method is not same as a teleoperation approach. Human hand movement cannot simply transferred to a robotic hand due to a difference in their structures. In addition, during an execution on a robotic hand, an open-loop control scheme is employed. No feedback system is used on the robot due to the nature of the proposed method and also limitations of resources.

1.3.1 Thesis Contributions

This thesis explains a method to teach a regrasping movement to a robotic hand. The contributions of the thesis can be described as the followings.

1. *A development of a methodology to allow a robot to imitate human regrasping movement and an introduction of a topological representation to use with human and robotic hands.* The method is based on LFO paradigm and uses a topological representation referred to as Gauss Linking Integral (GLI). The representation allows a human regrasping movement to be segmented and recognised. When the movement is reproduced using this representation, it also allows the movement to be efficiently interpolated in the topological space. The works in this thesis are believed to be the first method to apply this representation for representing and controlling human and robotic hands.

2. *A development of a method to segment human regrasping movement into meaningful segments based on its topological features.* The regrasping movement are segmented into a sequence of contact relation by considering changes in a type of writhe matrices. A writhe matric is a representation derived from the GLI. Two types of writhe matrices, peak-wm and span-wm, are introduced for this purpose. (Explanation of the content can be found in Chapter 4)

3. *A development of a task model to represent human regrasping movements and a method to recognise it.* The task model is based on types of writhe matrices. Three movement primitives, *detaching-attaching-crossover*, are

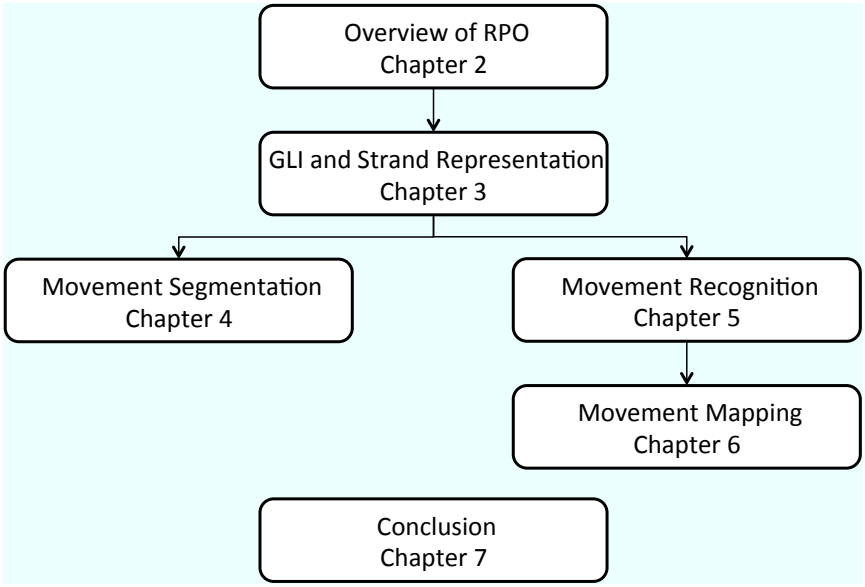


Figure 1.1: Diagram shows the organization of the thesis.

defined and used to represent the regrasping movement. A necessary skill parameters for mapping each movement primitive are defined and a method to extract them from human regrasping movement is described. (Explanation of the content can be found in Chapter 5)

4. *A development of a mapping framework to map a recognised movement to a robotic hand.* The framework comprised of two steps: mapping skill parameters to a robotic hand and interpolating between two robotic hand postures. A method to interpolate between robotic hand postures in a topological space is described. Human regrasping movement is reproduced and executed in a custom-made robotic hand to test and verify the proposed methodology. (Explanation of the content can be found in Chapter 6)

1.4 Organization of Thesis

Figure 1.1 shows a diagram of how the content are related to each other. The remainder of this thesis is organized as the followings:

In Chapter 2, a LFO paradigm is described. Based on the literatures, a method to design a system using LFO paradigm is explained. An overview of the proposed system is also given in the chapter.

Chapter 3 begins with an explanation on Gauss Linking Integral (GLI). A visualization of GLI is given together with how to calculate it, with the purpose of enhancing an understanding on the topic. Writhe matrix is also introduced in the chapter. It is a topological representation that is used throughout the

proposed method. The chapter ends with an explanation of how to represent hands and object as strands.

In Chapter 4, human regrasping movement is analysed with various features in a topological representation. Two method to segment the movement is explained. A classification of writhe matrice into two types is also explained here. The chapter is concluded with experimental results of both methods and their comparison.

Chapter 5 describes a task model used in the system and how to recognise it. A task model is the main component of the proposed system. The task model dictates what is important and to be recognised in human regrasping movement. The proposed task model is build upon a classification of writhe matrices. A grasp taxonomy to classify grasping postures is described. This leads to the definition of the task model: movement primitives and skill parameters. Then a method to recognise movement primitives and extract skill parameters from human regrasping movement is explained. The chapter concludes with experimental results of a recognition of regrasping movements.

In Chapter 6, a method to map a regrasping movement to a robotic hand is illustrated. The chapter begins with a method to refine skill parameters obtained from human demonstration. These skill parameters cannot be used directly to map the movement to a robotic hand. They are refined based on several constraints that would allow them to successfully generate a movement that can be used on the robotic hand. After that, a framework to map the movement to a robotic hand is explained. The framework describes how to generate a movement for each movement primitive in the topological space. It uses the knowledge extracted from human demonstration. The chapter then concludes with experimental results of mapping an example demonstrated movement onto a robotic hand.

Chapter 7 concludes the study in this thesis. The discussion describes an advantage of the proposed method. Future directions to improve and expand the method is given at the end of the chapter.

CHAPTER

TWO

Regrasping Planning from Observation (RPO)

Contents

In this chapter, an overview of proposed system is explained. An objective of Regrasping Planning from Observation (RPO) system is to allow a robot to imitate a regrasping movement shown by human. The design of the system is based on LFO paradigm. The key idea is to enable a system to observe human performing a regrasping movement, understand it, and duplicate the movement as similar as possible.

This chapter begins by explaining LFO paradigm, and how to design a system using LFO. Then, the chapter is concluded with an outline of the proposed RPO system.

2.1 Learning from Observation (LFO)

LFO, or sometimes refers to as Programming by Demonstration (PbD), is a paradigm to teach a robot to imitate various tasks from human. Human performs a desired task in front of a robot. The robot observes the task through many sensors, and try to recognise it. Once the robot obtains all the required knowledges, it performs the task on its own with minimum human intervention. There are various techniques and has been applied in many robotic applications. In general, these techniques can be divided into two categories [Billard *et al.* 2008]: imitation on trajectory level and imitation on symbolic level.

Imitation on trajectory level can be considered as an advanced version of teleoperation. In traditional teleoperation, trajectory of human movement is

captured and directly played back on the robot. Movement can be represented in various forms, e.g. a joint space or a position and orientation of an end effector, depending on a purpose of the movement. However, this direct playback approach has a drawback that the captured trajectory is often redundant. Some part of the movement is not really necessary to duplicate the original task. In conventional teleoperation, or LFO on trajectory level, the captured trajectory is modified before playing back on a robot. Various techniques, mainly are based on statistic, are proposed and applied on a captured trajectory, in order to retain only an essential information enough to reproduce an original task.

Imitation on symbolic level, on the other hand, uses a different approach to imitate a task. A model use to represent a task is first defined. This model contains an information of what is a necessary information to duplicate the task. A robot then recognises these informations from the captured trajectory to decide which task and how to perform the task. The model of a task can either be defined manually from human, or be extracted from multiple demonstrations of the task.

In this thesis, an imitation on symbolic level is used to teach robot a regrasping task. Specifically, it adopts an approach that was used to teach an assembly plan to a robot [Ikeuchi & Suehiro 1994]. The difference from other approaches is that it focuses on a particular domain of task, rather than a specific task. The model for imitation will be defined for the task domain. An advantage is that the defined model is more general, or in other word, the model is applicable for all movement in the task domain, not only to a specific movement.

2.1.1 Designing LFO System

To design a system using LFO paradigm, the most crucial component is a model for imitating a task. It is sometimes referred to as a *Task Model*. Two fundamental questions are needed to be answered in order to define a task model.

1. **What to Do?** What is an abstract representation that uses to represent a task, and how to recognise it? An abstract representation is usually described as a set of movement, often referred to as *movement primitive*.

2. **How to Do?** How to map each movement primitive to a robot, and what are the necessary parameters to describe each movement primitive? These parameters are usually referred to as *skill parameters*.

Teaching a task to a robot using LFO is divided into three phases: observation, recognition, and mapping. In observation phase, robot observes human performing a task through a set of sensors. During recognition phase, robot recognises human demonstration into a sequence of movement primitives, which in turn represents an original task. In mapping phase, the robot duplicate a

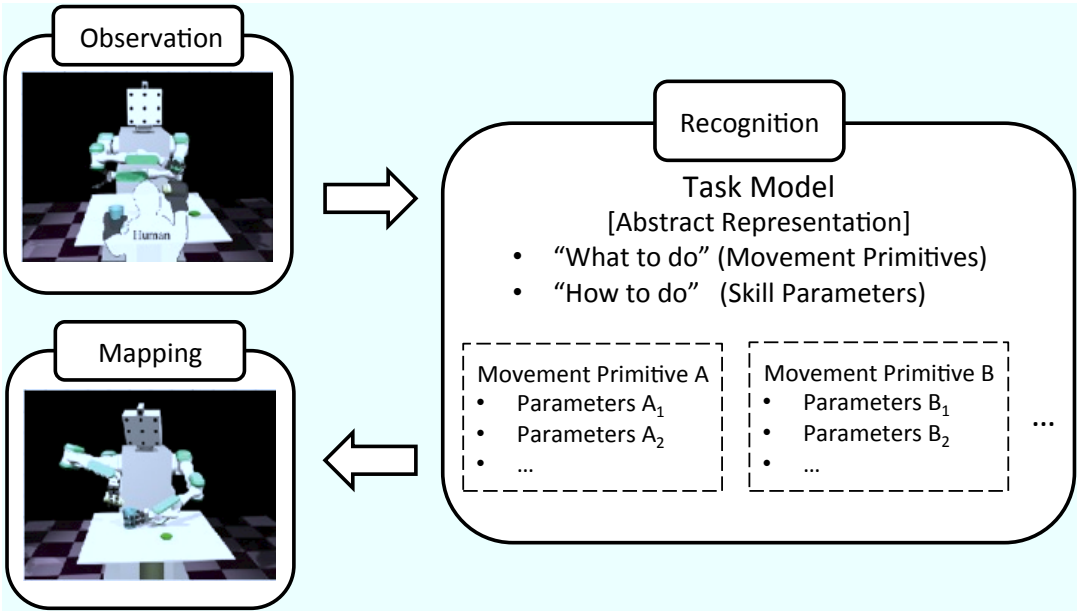


Figure 2.1: LFO paradigm consists of three important phases: observation, recognition, and mapping. Task model is an important element for a robot to recognise and understand a demonstration from human. It consists of movement primitives and a set of skill parameters for each movement primitive.

task by performing each movement primitive sequentially. The skill parameters extracted from human demonstration is used as an initial clue for the robot to perform each movement primitive.

An abstract representation, or a set of movement primitives, is a vital component for any LFO system. It bridges observation phase and mapping phase together. It is also used as a main element during recognition phase. Movement primitive is considered as a symbolic representation which is abstracted from both human and robot. By referring to abstraction, it means that a movement primitive performs on any human or robot is considered identical regardless of their structure.

In order to define a set of movement primitives for a specific task domain, a task or a movement must be characterised and classify into recognisable states using some criteria. A transition between these states will then be classified or grouped into one type of movement primitive, depending the nature of that particular problem.

Assembly Plan from Observation (APO) was one of the pioneer using this approach to design LFO system [Ikeuchi & Suehiro 1994, Suehiro & Ikeuchi 1992]. A system to teach an assembly task to a robot was developed. Assembly task is classified into various states based on their face contact relation. A total of thirteen movement primitives connect these states together. When human demon-

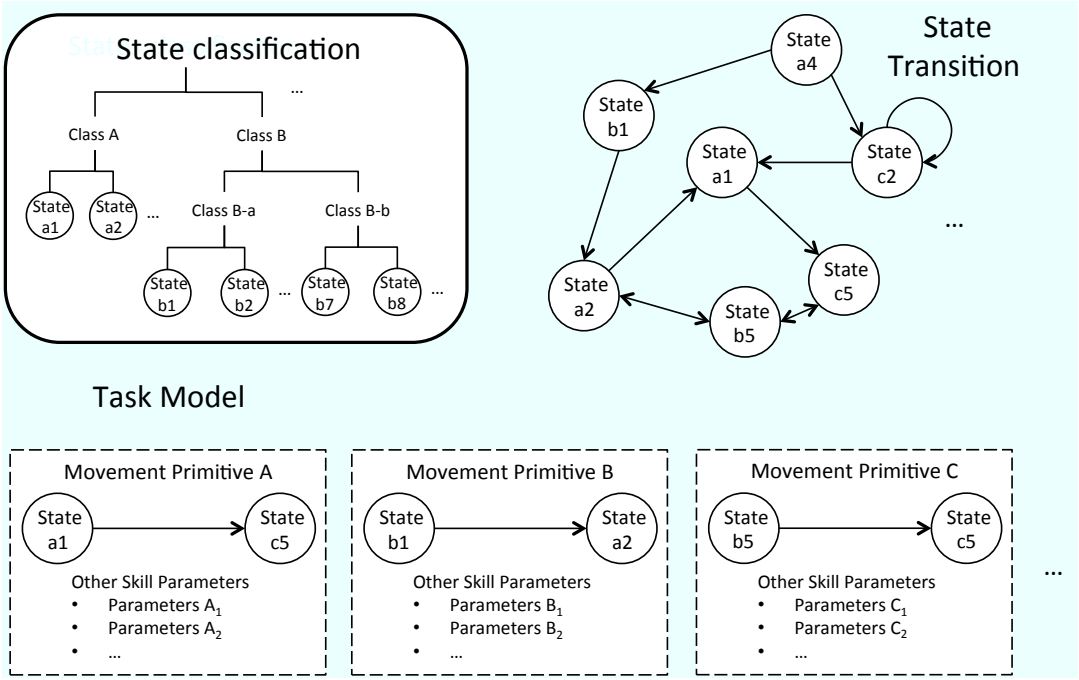


Figure 2.2: In a LFO paradigm, a specific task domain is represented and characterised by states, which are classified into groups by some criteria. State transition is a finite state machine that represents possibles changes. Movement primitives of a task model can be defined by (grouping) these transitions.

strates assembly task in front of a TV camera, movement primitive together with corresponding skill parameters are recovered from an image sequence by the recognition module.

An enhanced version of APO system was developed to deal with rotation movement in assembly plan [Takamatsu *et al.* 2007]. Contact relation between object was used as a representation; however, it is not used to classify a task because there would be an infinite number of states when any contact relations other than face contact relation are considered. States are characterised by types of legal infinitesimal displacement of an object, referred to as *motion*-DOF in that particular object configuration. Then, a set of movement primitives are defined based on changes of types of displacement.

Knot Planning from Observation (KPO) used the paradigm to construct a system to teach knot tying task [Takamatsu *et al.* 2006]. A P-Data representation is used to describe and characterise knot states. Although there is an infinite number of knot states classified by this P-Data representation, a set of four movement primitives is defined on this representation. This is because based on a knot theory, only four movements are required to tie any types of knot for a pair of opened strings.

LFO paradigm is also applied to teach a humanoid robot to imitate dance [Nakaoka *et al.* 2007]. A leg task model is defined to allow a biped robot to imitate a lower body movement of a Japanese dance. States in the system are classified by a waist location and contact relation between each leg and a floor. Transitions between these states are then group into four types of movement primitives.

2.2 Outline of RPO System

RPO is designed to teach a robot to imitate regrasping movement from human. It follows the LFO paradigm explained in Section 2.1. The teaching process can be divided into three phases: observation, recognition, and mapping. In an observation phase, robot observes a regrasping movement from human demonstration through a data acquisition system. Robot, then tried to understand and recognise an important information based on a pre-defined model in a recognition phase. Finally, in a mapping phase, robot uses a recognised information to reproduce a regrasping movement on its own hand. Figure 2.3 illustrates the outline of RPO system.

A data acquisition system observes a human regrasping movement through a motion tracker and a data glove. Data captured at a particular moment are a configuration of human hand, position and orientation of both human hand and a manipulated object. These data are referred to as a grasping posture. An observed regrasping movement, hence, is a sequence of grasping postures.

In a RPO system, task model is designed based on tangle topology. Tangle topology is a geometric property that describes the relation between two strands. Hand and the manipulated object are considered as zero-width strands. This allows each grasping posture to be considered as a tangle relation between strands representing hand and object. Contact relation between hand and object can then be realized from this tangle relation. Taxonomy of grasping postures is constructed based on contact relations, which will, in turn, use to classify states in the regrasping movement. Three movement primitives, a main element of the task model, are defined as movement that transfers a grasping posture from one state to another.

To simplify a complexity of the problem, a current task model is designed for a regrasping movement of a pen-like object (e.g. pen, paintbrush). It should, however, be applicable with any object that could be represented with one strand. Designing a task model of regrasping movement for an arbitrary object is our final goal, but this raises various difficult issues, such as how the object should be represented by strands, or how to map the movement primitives that interacts with more than one strands at the time.

An observed regrasping movement is segmented into many shorter sequences. They are then recognised into states of grasping posture. The pre-defined task model and these states information allow movement primitives to be recognised.

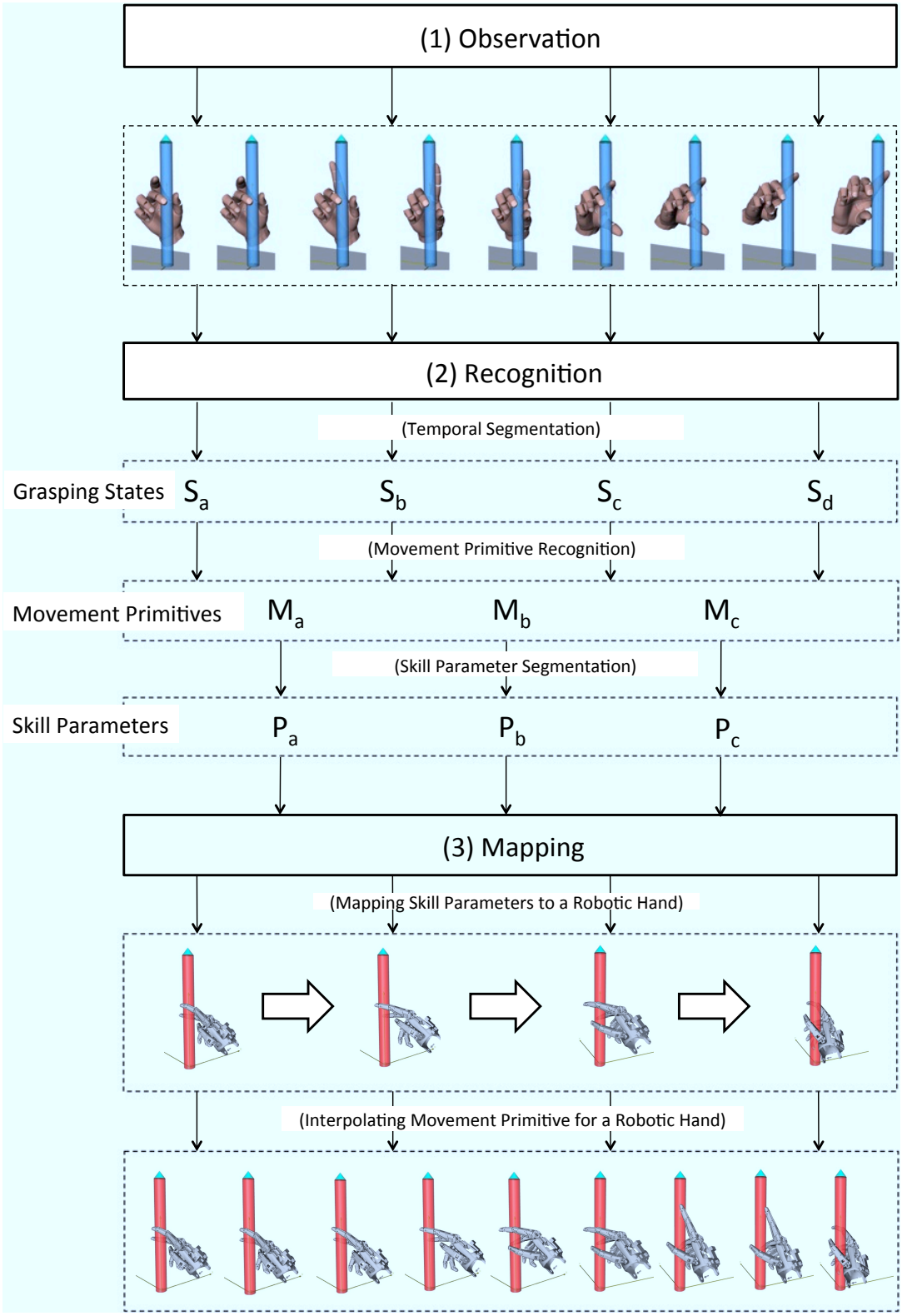


Figure 2.3: Outline of RPO System

An original regrasping movement is thus then be represented as a sequence of movement primitives. Once regrasping movement is represented as a sequence of movement primitives, skill parameters are used to describe how to reproduce each movement primitives. A set of required parameters is different depending on a type of movement primitives. These parameters are extracted directly from the observed movement. A small modification of these parameters is applied before reproducing each movement primitive. The objective of this parameter refinement is to make these parameters obeyed the physical constraints in the target environment.

A regrasping movement is duplicated on a target robot hand to verify the proposed task model. In a mapping phase, movement primitives that represent an original regrasping movement are mapped onto a robot hand sequentially. Framework to map movement primitive to a robot hand are presented. It is based on the interpolation in topological space. Each type of movement primitives are interpolated differently depending on its characteristic.

CHAPTER

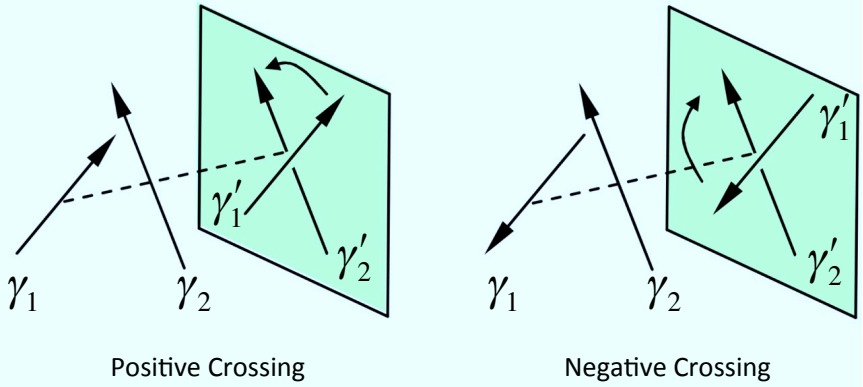
THREE

Representation of Hands and Object

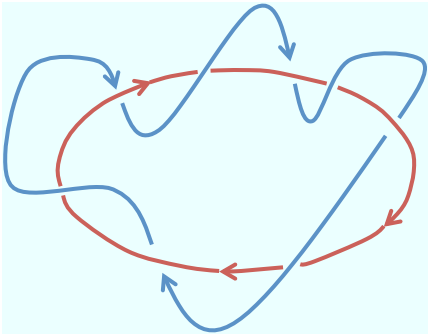
Contents

Gauss Linking Integral (GLI) is a topological representation that used throughout the system. It is very important that it is explained properly in details. GLI represents a geometric property between two parametric curves. However, for any curves that are represented by a sequence of line segments, they are calculated differently using the analytical approach. A writhe matrix is a by-product of the calculation. To use the representation on regrasping movement, the fundamental idea is to consider the hand and manipulated object as strands. This allows a grasping posture to be considered as a tangle relation between strands of hand and an object. Consequently, regrasping which is an alteration of grasping postures over time can be examined as a sequential altering of the tangle relation of the strands.

In this chapter, GLI is described together with its visualisation. Then one of the analytical approach to calculate GLI is explained. The chapter is concluded with an explanation of how hands, both human and robotic, and object are represented as strands, together with some examples.



(a) Signed crossing between two line segments [Au & Woo 2004].



(b) With 0 positive crossings and 8 negative crossing, these curves have a linking number of -4.

Figure 3.1: Signed crossing and linking number between two closed curves.

3.1 Gauss Linking Integral (GLI)

Linking number of two closed curves describes the number of times that each curve tangles around the other in three dimension space. This number is invariant to the viewing direction. A straightforward method to calculate a linking number of two curves can be done by projecting the two curves into a chosen plane and summing up their signed crossings.

Figure 3.1(a) illustrates two types of crossing between two line segments: a positive crossing and a negative crossing. Considering the segment on top, a crossing is positive if the angle($< \pi$) is required to rotate its arrow onto the arrow of the bottom segment is counter-clockwise; otherwise, it is negative [Au & Woo 2004]. Linking number is equal to half of a sum of all signed crossings between two curves.

Linking number can also be calculated using Gauss Linking Integral (GLI) for two mathematically-defined curves. Let γ_1 and γ_2 represent two curves, GLI

of the two curves is computed by the following equation :

$$GLI(\gamma_1, \gamma_2) = \frac{1}{4\pi} \int_{\gamma_1} \int_{\gamma_2} \frac{d\gamma_1 \times d\gamma_2 \cdot (\gamma_2 - \gamma_1)}{\|\gamma_2 - \gamma_1\|^3}. \tag{3.1}$$

In short, it is a number of average crossing between two curves when viewing from all viewing directions in three dimensional space. Note that Gauss Linking Integral (GLI) of two zero-width curves can be directly used to describe a linking number of the two curves; however, this is not the case when a ribbon is considered [Dennis & Hannay 2005].

3.1.1 Visualisation of GLI

To understand the geometric interpretation of GLI equation, the concept of crossing between two line segments is first explained. Considering a point γ_1 on a line segments $d\gamma_1$ and a point γ_2 on a line segments $d\gamma_2$, two line segments are crossed if and only if there is a *line of sight* $\overrightarrow{\gamma_1\gamma_2}$ connecting between the two points [Au 2008].

All line of sights between any two points on line segments, $d\gamma_1$ and $d\gamma_2$, can be combined into an angle $d\Omega$ in a three dimensional space. Considering a segments $d\gamma_1$ as a source of viewing, points $\gamma_{m,i}$ and $\gamma_{m,j}$ as the starting and ending points of a segment $d\gamma_m$, all line of sights from a point $\gamma_{1,i}$ to segments $d\gamma_2$ are described by triangle $\triangle\gamma_{1,i}\gamma_{2,i}\gamma_{2,j}$. Similarly, all line of sights from a point $\gamma_{1,j}$ to segments $d\gamma_2$ are described by triangle $\triangle\gamma_{1,j}\gamma_{2,i}\gamma_{2,j}$. $d\Omega$ is then an angle at the top of a quadrangular pyramid that connects the two triangle together by moving $\gamma_{1,i}$ to $\gamma_{1,j}$ as shown in Figure 3.2. Note that $\triangle\gamma_{1,i}\gamma_{2,i}\gamma_{2,j}$ becomes $\triangle\gamma_{1,j}\gamma'_{2,i}\gamma'_{2,j}$ once moved.

The angle $d\Omega$ at the top of the quadrangular pyramid $\gamma_{1,j}$ can be formally defined as a solid angle. Considering the quadrangular pyramid with $\gamma'_{2,i}\gamma_{2,i}\gamma_{2,j}\gamma'_{2,j}$ as a quadrangular base, an area of the quadrangular is equal to $dA' = d\gamma_1 \times d\gamma_2$. Adjusting the quadrangle so that it is perpendicular to a *line of sight* $\overrightarrow{\gamma_1\gamma_2} = \gamma_2 - \gamma_1$, its area can now be expressed as

$$dA = \frac{d\gamma_1 \times d\gamma_2 \cdot (\gamma_2 - \gamma_1)}{\|\gamma_2 - \gamma_1\|}. \tag{3.2}$$

The solid angle $d\Omega$ that represents all *line of sights* between two line segments, $d\gamma_1$ and $d\gamma_2$, is given by

$$d\Omega = \frac{dA}{\|\gamma_2 - \gamma_1\|^2} = \frac{d\gamma_1 \times d\gamma_2 \cdot (\gamma_2 - \gamma_1)}{\|\gamma_2 - \gamma_1\|^3}, \tag{3.3}$$

where $\|\gamma_2 - \gamma_1\|^2$ normalises the area onto a unit sphere.

GLI in Equation (3.1) is simply a summation of all solid angle between two

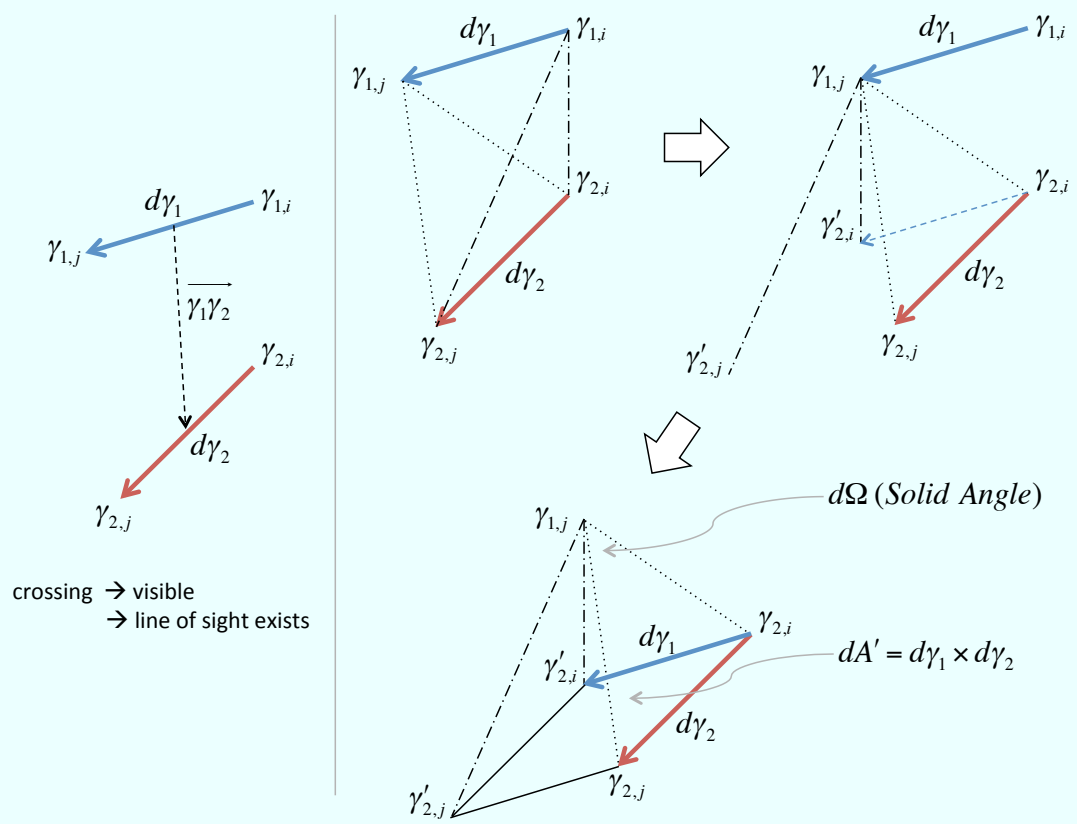


Figure 3.2: *Line of sight* between two line segments. The combination of these lines of sight can be visualized as a quadrangular pyramid.

curve segments, where the number of crossing is averaged by dividing with 4π which is a total solid angle in a unit sphere.

3.1.2 Calculation of GLI

GLI cannot be calculated directly from Equation (3.1), when curves cannot be represented by a closed form function, but a chain of line segments. Consider two curves S_1 and S_2, each curve is represented by line segments of n_1 and n_2 segments respectively. GLI of the two curves can be calculated by a summation of a GLI of every pair of line segments. Klenin and Langowski proposed four analytical solutions for calculating a GLI of two line segments [Klenin & Langowski 2000]. One of the solution which is based on pure geometry is described in this subsection. In short, it described a method to calculate the solid angle between two line segments explained in Section 3.1.1.

For simplicity, a notation is redefined. Consider a line segment (with direction) $\mathbf{r}_{ab}$ with points a and b as its starting and ending point, and a line segment (with direction) $\mathbf{r}_{cd}$ with points c and d as its staring and ending point. A quad-

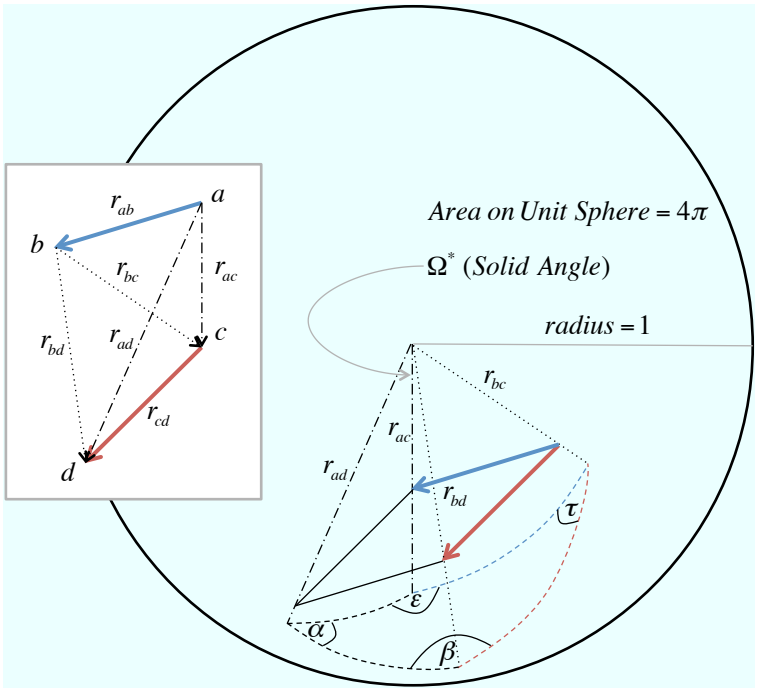


Figure 3.3: A quadrangular base of the pyramid is projected onto a unit sphere by extending the rays that connect to the apex. A solid angle Ω^* can be calculated by angles of the quadrangular base on the sphere; $\Omega^* = \alpha + \beta + \tau + \epsilon - 2\pi$.

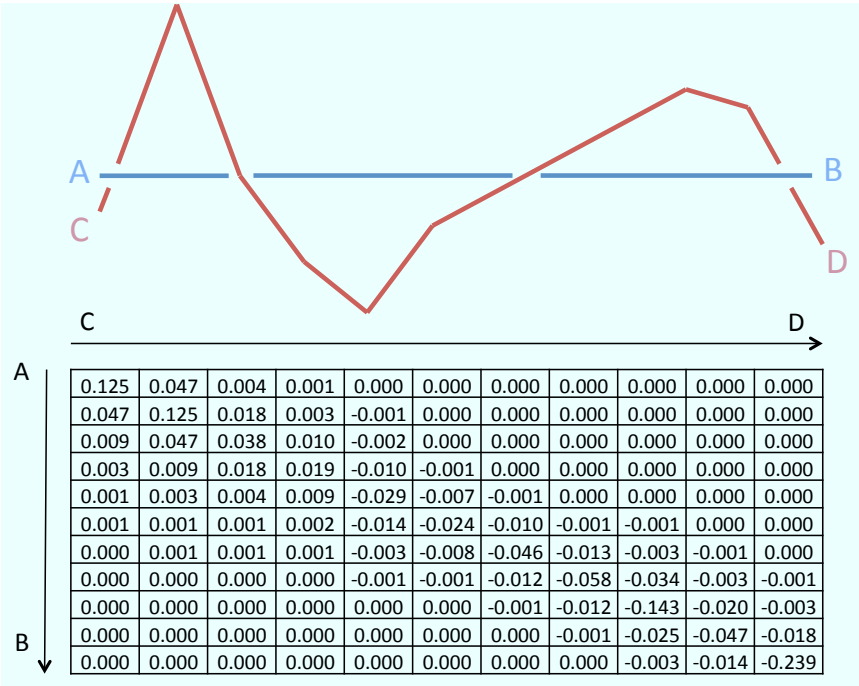
rangular pyramid can be constructed using a concept of line of sight as described in Section 3.1.1. GLI of these two line segments are defined as $\Omega/4\pi$, where Ω is a solid angle of the quadrangular pyramid. A solid angle is defined on a unit sphere. To calculate Ω, the quadrangular base of the pyramid is projected onto a unit sphere by projecting rays that connected to the apex of the pyramid to create a new quadrangular base as shown in Figure 3.3. The Ω^*, or in other word the area (positive) of this quadrangle on a united sphere, can then be calculated by

$$\Omega^* = \alpha + \beta + \tau + \epsilon - 2\pi, \tag{3.4}$$

where $\alpha, \beta, \tau, \epsilon$ are angles of the quadrangle base on a solid sphere.

Giving that $\mathbf{r}_{ac}, \mathbf{r}_{ad}, \mathbf{r}_{bc}, \mathbf{r}_{bd}$ is a vector connecting points a-c, a-d, b-c, b-d respectively, unit vectors normal to the planes bounding the pyramid can be described as

$$\begin{aligned} \mathbf{n}_a &= \frac{\mathbf{r}_{ac} \times \mathbf{r}_{ad}}{\|\mathbf{r}_{ac} \times \mathbf{r}_{ad}\|}, & \mathbf{n}_b &= \frac{\mathbf{r}_{ad} \times \mathbf{r}_{bd}}{\|\mathbf{r}_{ad} \times \mathbf{r}_{bd}\|}, \\ \mathbf{n}_c &= \frac{\mathbf{r}_{bd} \times \mathbf{r}_{bc}}{\|\mathbf{r}_{bd} \times \mathbf{r}_{bc}\|}, & \mathbf{n}_d &= \frac{\mathbf{r}_{bc} \times \mathbf{r}_{ac}}{\|\mathbf{r}_{bc} \times \mathbf{r}_{ac}\|}. \end{aligned} \tag{3.5}$$



0.125	0.047	0.004	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.047	0.125	0.018	0.003	-0.001	0.000	0.000	0.000	0.000	0.000	0.000
0.009	0.047	0.038	0.010	-0.002	0.000	0.000	0.000	0.000	0.000	0.000
0.003	0.009	0.018	0.019	-0.010	-0.001	0.000	0.000	0.000	0.000	0.000
0.001	0.003	0.004	0.009	-0.029	-0.007	-0.001	0.000	0.000	0.000	0.000
0.001	0.001	0.001	0.002	-0.014	-0.024	-0.010	-0.001	-0.001	0.000	0.000
0.000	0.001	0.001	0.001	-0.003	-0.008	-0.046	-0.013	-0.003	-0.001	0.000
0.000	0.000	0.000	0.000	-0.001	-0.001	-0.012	-0.058	-0.034	-0.003	-0.001
0.000	0.000	0.000	0.000	0.000	0.000	-0.001	-0.012	-0.143	-0.020	-0.003
0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.001	-0.025	-0.047	-0.018
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.003	-0.014	-0.239

Figure 3.4: An example of a calculation of a writhe matrix of a pair of curves. A total writhe of the writhe matrix is equal to -0.2630.

Angles of the quadrangle base can then be expressed as, $\alpha = \arccos(-\mathbf{n}_a\mathbf{n}_b) = \pi/2 + \arcsin(\mathbf{n}_a\mathbf{n}_b)$, $\beta = \pi/2 + \arcsin(\mathbf{n}_b\mathbf{n}_c)$, $\tau = \pi/2 + \arcsin(\mathbf{n}_c\mathbf{n}_d)$, and $\epsilon = \pi/2 + \arcsin(\mathbf{n}_d\mathbf{n}_a)$. This results in GLI of the two line segments as,

$$\begin{aligned}\frac{\Omega}{4\pi} &= \frac{k\Omega^*}{4\pi} \\ &= \frac{k}{4\pi}\left(\arcsin(\mathbf{n}_a\mathbf{n}_b) + \arcsin(\mathbf{n}_b\mathbf{n}_c) + \arcsin(\mathbf{n}_c\mathbf{n}_d) + \arcsin(\mathbf{n}_d\mathbf{n}_a)\right), \end{aligned} \quad (3.6)$$

where $k = 1 * \text{sign}((\mathbf{r}_{cd} \times \mathbf{r}_{ab}) \cdot \mathbf{r}_{ac})$ indicates sign of GLI value.

3.1.2.1 Writhe Matrix

A **writhe matrix** [Ho & Komura 2009a] is a by-product from the calculation of GLI between two curves that are represented as a chain of line segments. Considering two curves S_1 and S_2, each curve is represented by a chain of line segments of n_1 and n_2 segments respectively. A writhe matrix (T) is a $n_1 \times n_2$ matrix whose element T_{ij} is the GLI between segment i of S_1 and segment j of S_2 which calculated using a method explained previously in the section. GLI between two curves, or sometimes referred to as **total writhe**, is a sum of the GLI of every pair of segment. Note that the terms *total writhe* and *writhe* might

be used interchangeably. Figure 3.4 shows an example of writhe matrix between two curves together with its total writhe value.

3.2 Representing Hands and Object as Strands

To analyse and imitate a human regrasping movement using tangle topology, the structure of a human hand, a robotic hand, and a pen must be represented as strands (or curves). Each strand is a zero-width strand which deforms accordingly to the object it represents. To start with, hands substituted with the same number of strand as the number of their fingers. Each strand is directly substituted for each finger. The pen is substituted with one straight strand. Without a loss of generality, direction of the strands of the hands are chosen to be from the tip of the fingers to their proximal joint and direction of the strand of the pen is chosen to be from the bottom to the tip of the pen. Figure 3.5(a) shows how hands and pen are substituted with strand structure.

An alternative method to represent the hands is shown in Figure 3.5(b). Hand are substituted with strands, where each strand is a connection between the tips of any two fingers. In case of a hand with five fingers, the hand is substituted with $\binom{5}{2} = 10$ strands. Similarly, the direction of the strands starts from the tip of the former to the latter finger.

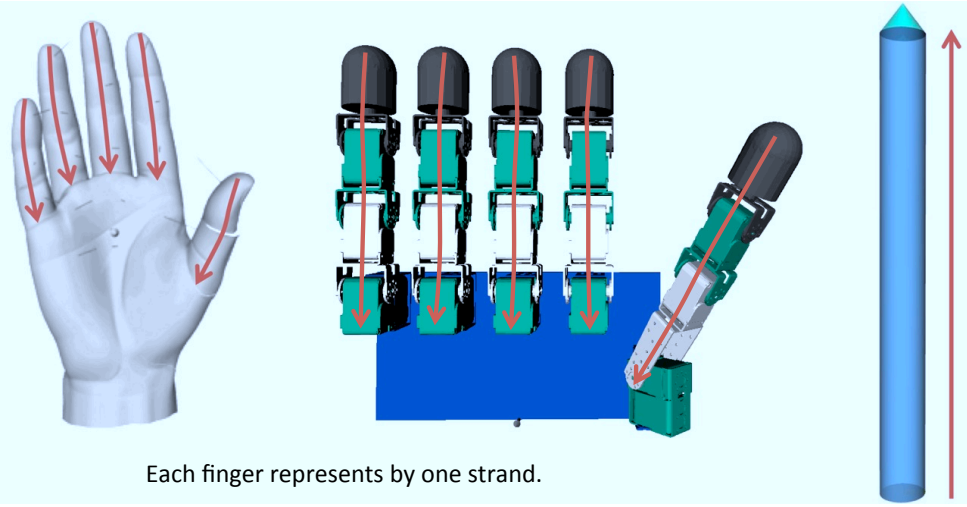
In general, the former representation is used throughout this research. However, the latter is used in some situations, which will refer to the representation as an *alternative representation* of hands.

3.2.1 Experimental Results

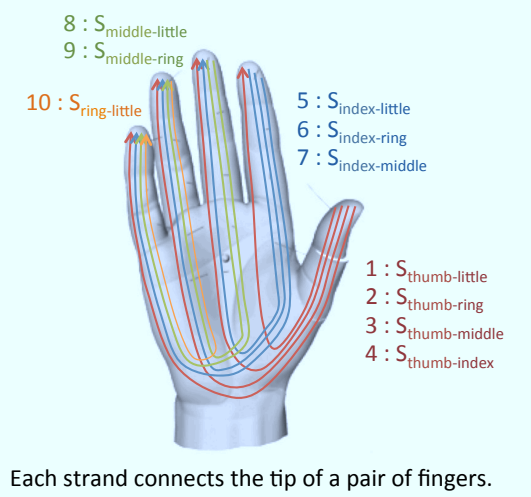
Writhe matrices between fingers and a pen are the fundamental features in a tangle topology used throughout the system. This section shows some calculated results of writhe matrices of grasping postures. Writhe matrices of a grasping posture of human hand is calculated, followed by a grasping posture of a robotic hand.

Figure 3.6 shows calculated writhe matrices of an example of a human grasping posture. Both strand representations of the hand are considered, and the corresponding write matrices of some fingers are calculated for comparison. It can be noticed that the writhe matrix shown in Figure 3.6(b) is a combination of thumb-pen and (minus of) index-pen writhe matrices shown in Figure 3.6(a). This is because both strand representations contains common line segments, e.g. strand that represents thumb-index finger has common line segments with strands that represent thumb and index finger.

Similarly, Figure 3.7 shows calculated writhe matrices of an example of a grasping posture of a robotic hand. This can be done easily because of an abstraction of a strand representation. In other words, there is no distinction

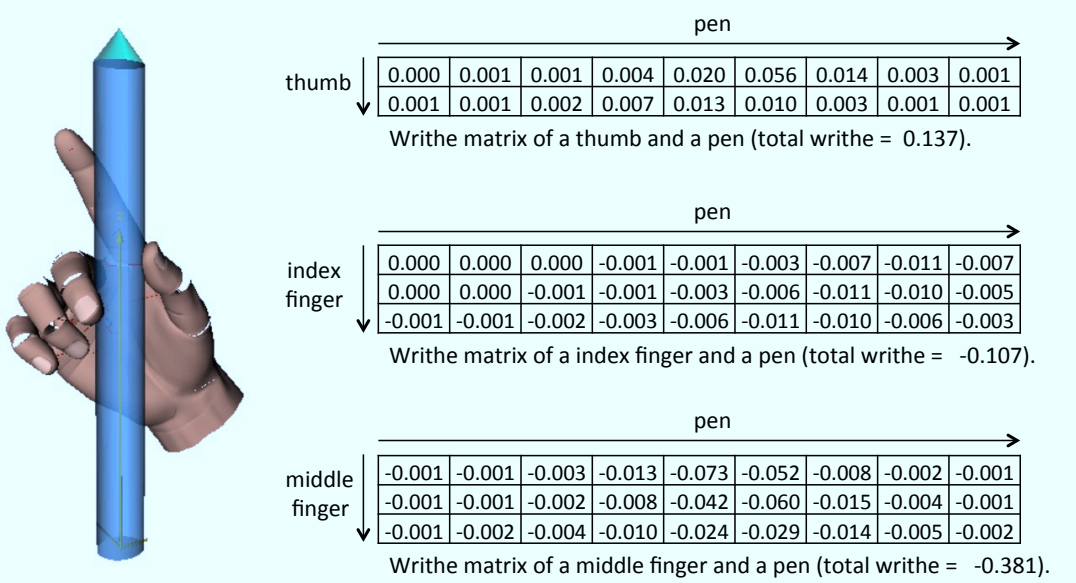


(a) Hands and a pen are represented with zero-width strands. For a hand, each finger is represented with a strand. Thus, five strands is needed a five-finger hands. One strand is used to represent a pen. The direction of strands effect the sign of the calculate writhes. Without a loss of generality, the direction of strands of the hands are from their tip to the corresponding proximal joint and the direction of the strand of a pen is from its bottom to its tip.

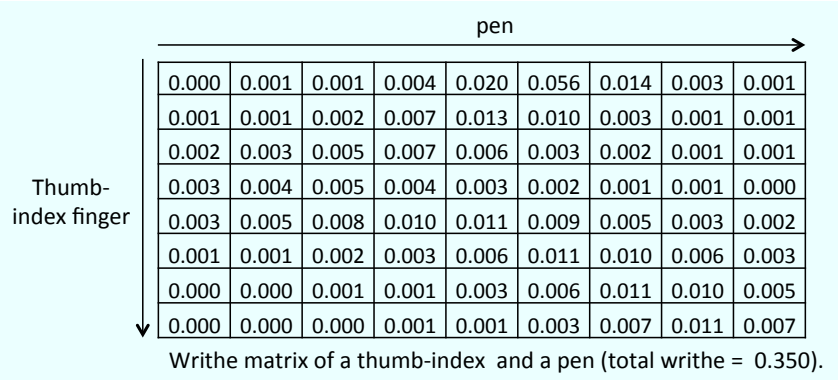


(b) An *alternative representation* of how to represent a hand. Each strand connect tips of two fingers. Human hand can be represented by a total of ten strands.

Figure 3.5: Representing hands and an object as strands.



(a) An example of a grasping posture of a human hand is given. Three writhe matrices of three pairs of strands (thumb-pen, index-pen, middle-pen) are calculated, together with their total writhe. It can be noticed that a sign of a total writhe of the thumb-pen writhe matrix is opposite to the others. This is because the thumb is on the different side of the pen, when comparing to the index and middle finger.



(b) Writhe matrix of a human grasping posture shown in Figure 3.6(a) is calculated, together with its total writhe. It is a writhe matrix of a pair of strands; one represents a thumb-index finger as illustrated in Figure 3.5(b), and the other represents a pen.

Figure 3.6: Writhe matrices of a grasping posture of a human hand.

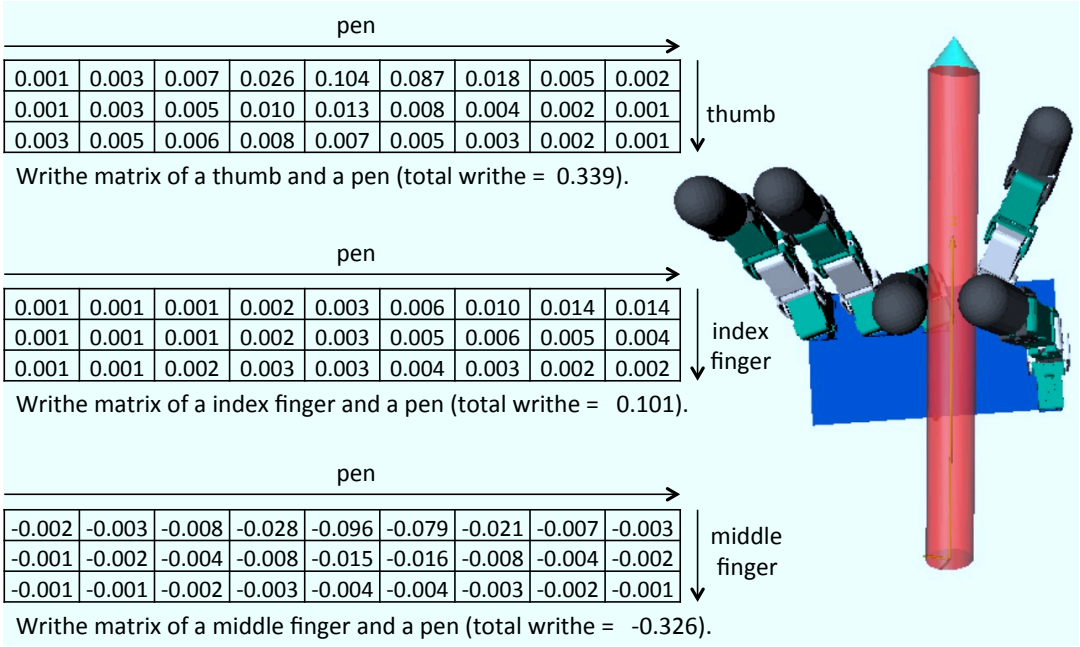


Figure 3.7: An example of a grasping posture of a robotic hand is given. Three writhe matrices of three pairs of strands (thumb-pen, index-pen, middle-pen) are calculated, together with their total writhe. It can be noticed that a sign of a total writhe of the middle-pen writhe matrix is opposite to the others. This is because the middle finger is on the different side of the pen, when comparing to the thumb and index finger.

between two types of hand, as long as they are represented as strands, they are treated the same in tangle topology.

CHAPTER

FOUR

Segmentation of Regrasping Movements

Contents

In this chapter, methods to segment regrasping movements are described. An objective of the segmentation is to divide a human movement into sequence of shorter movements, which will later be classified into more meaningful representation. A tangle relation of hand and object during regrasping are analysed. Methods to segment a regrasping movement based on features in tangle topology are proposed and compared.

The chapter starts by reviewing literatures on human manipulative movement. Then features in topology space for analysing regrasping movement are described, followed by the proposed segmentation methods. The chapter concludes with the experimental results and their comparison.

4.1 Studies on Human Prehensile Movements and its Segmentation

Human prehensile movement has been widely studied in a medical field. Most of which focus on classifying and categorising types of *static* human grasps. More detail review on these classifications is given in Section 5.1.1. However, there is only one classification to provide clear definition on types of *dynamic* hand prehensile movements. Elliott and Connolly classified dynamic human prehensile movements (referred to as intrinsic movements in the paper) as either simultaneous or sequential movement patterns [Elliott & K.J. 1984]. Movements are a simultaneous movement when all fingers moves in a co-ordinated pattern, while fingers move independently depending on its roles in a sequential movement.

In general, most methods analyse and segment human prehensile movement based on changes of some particular features. The movement is segments into shorter movements, where each of them will be recognised into meaningful states based on some classification or pre-defined model. Features used to analyse and segment the movement can be broadly divided into two groups: features that are derived from the trajectory of the hand based on a particular characteristic or relation, and features as a contact relation.

Kang and Ikeuchi used hand fingertip polygon and hand volume sweep rate to temporally segment hand movement into pregrasp, grasp and manipulation phase [Kang & Ikeuchi 1995]. However, they considered only a *homogeneous* manipulation which is not involved in a change of fingertip or grasp type. Zacksenhouse and Moestl proposed a method to segment human manipulative movement based on phase-plane [Zacksenhouse & Moestl 1999]. Two most active joints define a phase-plane for each movement. The segmentation is conducted based on the assumption that the trajectories of simultaneous movement are linear due to the coordination of both joints. However, for sequential movements or complex movements, the method would require multiple phase-planes as a segmentation feature.

Dejmal and Zacksenhouse extended the idea of the correlation structure of joints during the manipulative movement [Dejmal & Zacksenhouse 2006]. They applied Principal Component Analysis (PCA) onto all joints of the hand during the repetition of simultaneous hand movement. It appeared that the 1st Principal Component (PC) accounts most of the variance and captures well the complete movement. The segmentation method is conducted the 1st PC when it reaches an extreme point. Vinayavekhin *et al.* also applied similar approaches on more complex movements [Vinayavekhin *et al.* 2009]. However, they used the first three of PC instead as suggested by the experimental results.

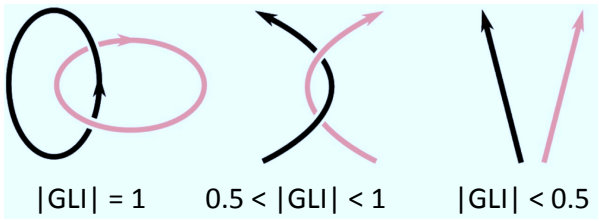
Using contact relation as a segmentation feature is a prominent choice, as most manipulative movements changes a contact between hand and object in order to change a grasp configuration. An obvious approach to observe the

changes of contact is to use a tactile sensor [Yousef *et al.* 2011, Johansson & Flanagan 2009]. Zollner *et al.* used an observed force value to distinguish and segment a prehensile movement into three different phases: static grasp, external forces, and dynamic grasps [Zollner *et al.* 2002]. Then a Support Vector Machine (SVM) classified were used to identify the type of each dynamic grasps. Bernardin *et al.* combined pressure value from the tactile sensor array and joint angle information as features for Hidden Markov Models (HMM) [Bernardin *et al.* 2005]. The proposed method segmented a human movement into shorter movement, which can be classified into one type of grasp. Kondo *et al.* attached a tactile sensor sheet on the object to capture pressure value during in-hand manipulation movement [Kondo *et al.* 2008]. A pre-defined contact state on the hand is discovered by processing a pressure distribution image. The movement is segmented into a transition of contact states using a dynamic programming. Similarly, Faria *et al.* segmented in-hand manipulative movement into a transition of pre-defined contact states [Faria *et al.* 2012]. However, the tactile sensors were attached directly onto human hand, and Bayesian model was used to continuous classify and segment the movement.

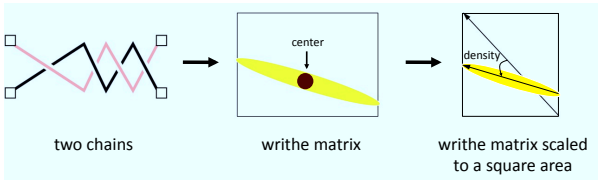
An alternative approach to observe contact relation is to demonstrate a movement in a virtual environment [Aleotti & Caselli 2006]. In general, a glove data and motion tracker are used to track hand configuration and location. The demonstration is then performed in the virtual environment, while a physic collision engine simulates and provides a precise contact information. Scharfe *et al.* developed the system for Shadow Hand model, and demonstrated in-hand manipulative movement of a wooden block [Scharfe *et al.* 2012]. The system shown a promising capturing results of the contact information; however, a method to segment and recognise the movement were not given.

4.2 Features for Segmentation

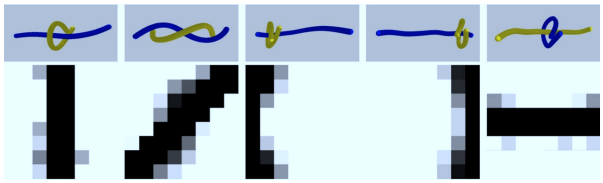
In this research, features in topology space are used to segment a human regrasping movement. Two sets of features are proposed based on writhe matrices between strands of fingers and object. Attributes of both sets of features are extracted from writhe matrices depending how they are perceived. The first set of features is called topological coordinate, and it models all writhe matrices identically. Topological coordinate consists of three attributes that describe each writhe matrix. On the other hand, the second set of features is based on the idea to distinguish writhe matrices into two types. The classification of writhe matrices depends on whether there is a contact relation between fingers and the object that constitute them.



(a) Writhe in various conditions of strands.



(b) Center and Density in Writhe Matrix.



(c) Density in various conditions of strands.

Figure 4.1: A topological coordinate of writhe matrix. [Ho & Komura 2009a].

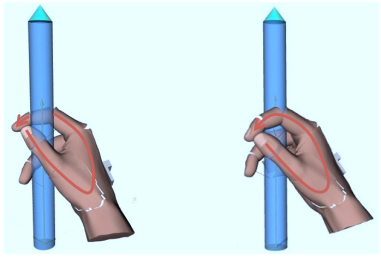
4.2.1 Writhe matrices as Topological Coordinate

Consider a writhe matrix T which is calculated from strand S_1 of size n_1 and strand S_2 of size n_2, a topological coordinate of the writhe matrix of size $n_1 \times n_2$ is consisted of three attributes [Ho & Komura 2009a]: **writhe** ($w \in \mathbb{R}$), **center** ($\mathbf{c} \in \mathbb{R}^2$), and **density** ($d \in \mathbb{R}$). The first attribute, w, is simply a sum of all elements of writhe matrix.

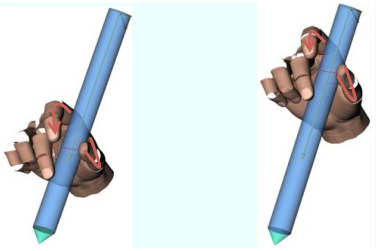
$$w = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} T_{i,j}. \tag{4.1}$$

The other attributes are introduces based on the characteristic of the writhe matrix. Center, a two dimensional vector (c_i, c_j), describes an overall location of the tangled area. The first element c_i indicates the position on strand S_1 where a center of the tangle formed by S_2 is, whereas c_j indicates the position on strand S_2 where the tangle formed by S_1 is centered. Center is calculated by computing the center of mass of all elements in writhe matrix as the following,

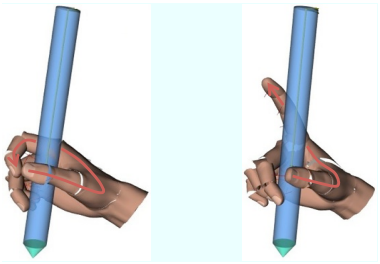
$$\mathbf{c} = (c_i, c_j) = \left(\frac{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} i \cdot T_{i,j}}{w} - \frac{n_1}{2}, \frac{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} j \cdot T_{i,j}}{w} - \frac{n_2}{2} \right). \tag{4.2}$$



(a) Grasping postures with different values of writhe. It can be noticed that a writhe of the left posture is close to zero.



(b) Grasping postures with different values of center (c_y which represents a location of the tangle on the pen).



(c) Grasping postures with different values of density.

Figure 4.2: Grasping postures with different topological coordinates when considering writhe matrices of thumb-index and pen.

Density describes how the two strands are tangled around each other. It indicates how much the tangle area is concentrated at one location along the strands. Density is defined by the angle made between the principle axis and the diagonal line as shown in Figure 4.1(b), once the writhe matrix is scaled to a square writhe matrix.

When S_1 and S_2 are strands representing fingers and a pen as described in Section 3.2 with the *alternative representation* of hands, each attribute of topological coordinate can be interpreted as the following. Writhe describes the number of tangle on the pen made by two fingers. Since the number of tangle is usually either 0 or 1 (and -1), the attribute indicates whether a particular pair of fingers is wrapping around the pen or not. Since the pen does not deform and wrap around the finger, c_i of center indicates the average location on the finger that tangles around the pen and c_j of center indicates the average location on the pen that is tangled by the finger, if any tangle exists. Finally, density indicates the orientation of the two fingers with the pen, if the tangle existed between them. Figure 4.2 illustrates pairs of grasping postures that have different topological coordinate. It emphasised the different of each attribute separately.

4.2.2 Writhe matrices as Peak-wm or Span-wm

The structure of a human hand and a manipulated object (pen-like object) are represented as strands described in Section 3.2. One strand from the hand and another strand from the object form a pair of strands. It is these pairs of strands whose writhe matrix will be analysed.

In general, writhe matrix could be in various shapes depending how the two strands are tangled together. Ho and Komura proposed a method to encode writhe matrices of motions of characters, and use it to retrieve similar motions [Ho & Komura 2009b]. However, in a very specific case where a finger can barely tangled around the object more than one round, writhe matrices could be examine differently.

Based on an observation and analysis of writhe matrices of various human manipulative movements, e.g. movements in the performance of Japanese tea ceremony, movements of artists during painting, writhe matrices of hand manipulative movement can be broadly classified into two types: peak-type writhe matrix (peak-wm) and span-type writhe matrix (span-wm).

- **Peak-wm** (T^p) is a writhe matrix whose majority of non-zero elements are concentrated at a specific area. This type of writhe matrices is used to represent grasping postures where the corresponding finger is in contact with the object.

- **Span-wm** (T^s), in contrast, is a writhe matrix whose non-zero elements are spread across the matrix. It represents grasping postures where the corresponding finger is located at some distance from the object.

Figure 4.3 illustrates an example for each types of writhe matrices, where the thumb finger of grasp posture in the figure are in contact with a pen and the index finger is not.

The classification of writhe matrices is also comprehensible from Equation (3.1). The denominator of GLI is (a power of three of) a distance between two curves or two line segments, when curves are represented with line segments. It greatly affects the value of the writhe. When a finger is in contact with a manipulated object, portions of line segments that represent the finger and the object are close together. This leads to higher writhe values in a specific area of the writhe matrix which is referred to as *peak* area.

For both types of writhe matrices, writhe is not only reflecting the average distance of the finger from the object, but also indicating the orientation of the finger to the object with its value and its sign. In Equation (3.1), each strand is assigned with the direction. Sign of writhe is determined by the numerator. Depending on the direction of strands of the finger and the object, writhe changes its sign.

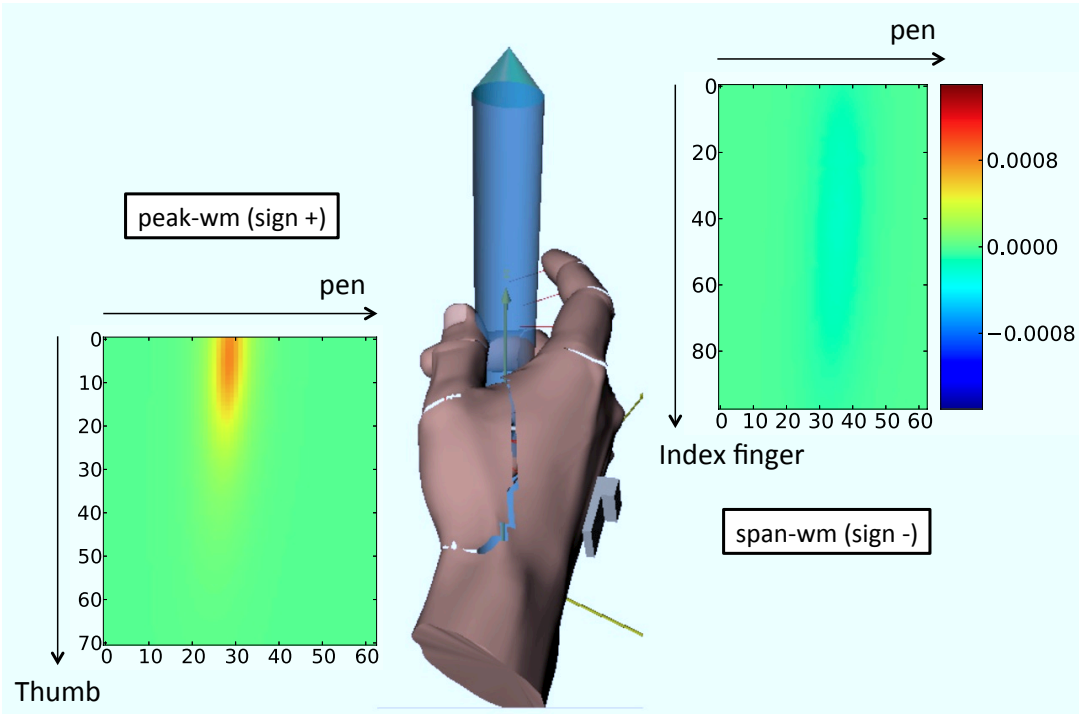


Figure 4.3: Example of two types of writhe matrices. Writhe matrix between thumb and pen is peak-wm (sign+), while writhe matrix between index finger and pen is span-wm (sign-).

In short, two types of writhe matrices distinguish whether there is a contact relation between the finger and the object. The segmentation method described later in the chapter is based on this concept.

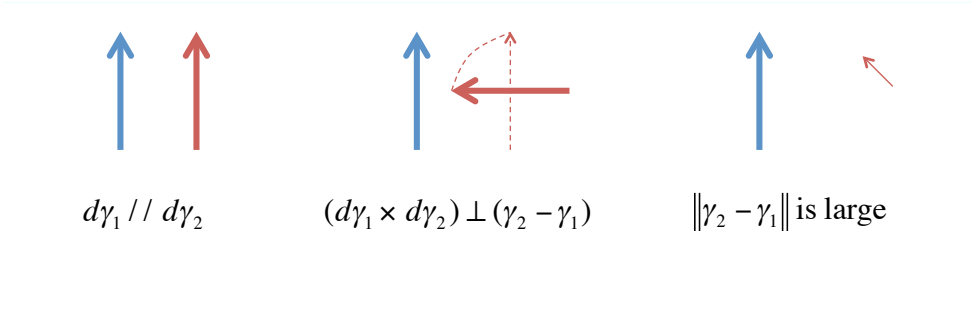
4.2.2.1 Exceptions : Writhe matrices as Zero-wm

It can be noticed that writhe matrix (or a GLI) is a representation that embeds a lot of information and is very rich in its own. However, there are some cases that it cannot represent well. To illustrate this, GLI equation is again examined. Consider two line segments $d\gamma_1, d\gamma_2$ and Equation (3.1), a writhe between the two segments is equal (or close) to zero in three situations:

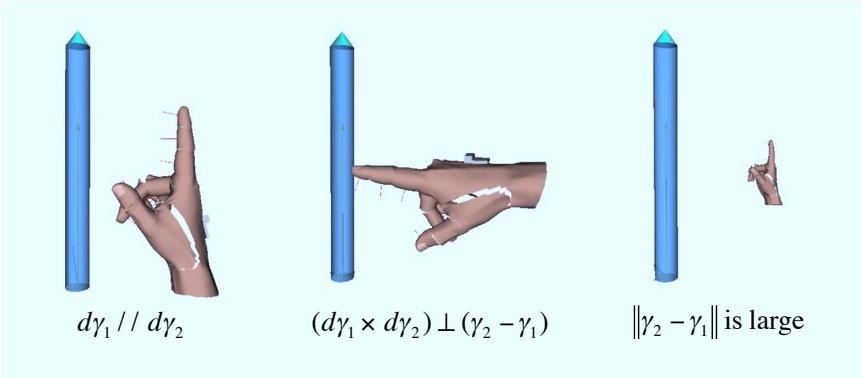
- when $d\gamma_1$ and $d\gamma_2$ are parallel to each other ($d\gamma_1 \times d\gamma_2 = 0$),
- when $d\gamma_1$ and $d\gamma_2$ are on the same plane ($d\gamma_1 \times d\gamma_2 \cdot (\gamma_2 - \gamma_1) = 0$),
- when $d\gamma_1$ and $d\gamma_2$ are far away from each other ($\|\gamma_2 - \gamma_1\|$ is large).

All three situations are shown in Figure 4.4(a).

When considering a writhe matrix calculated from a grasping posture, it is very rare that a finger and a pen would be in the orientation that all elements of



(a) Special situations of two line segments where their writhe is equal (or close) to zero.



(b) Writhe matrices between the index finger and the pen of these grasping postures are classified to be zero-wm.

Figure 4.4: Zero-wm: a special case of span-wm.

their corresponding writhe matrix would be all zeros. However, there are specific cases that most elements of the writhe matrix would be close to zero, which result in its total writhe to be close to zero as well. In this specific case, the writhe matrix is referred to as a zero writhe matrix (zero-wm), a special kind of span-wm. Grasping postures whose writhe matrices between index finger and a pen is considered to be zero-wms are illustrated in Figure 4.4(b).

4.3 Temporal Segmentation of Regrasping Movement

A data acquisition system captures a regrasping movement into a time series of grasping postures. Writhe matrices of all grasping postures in the sequence are calculated for all pairs of strands depending on the chosen representation of the hand. The regrasping movement is then segmented into many shorter movements based on some features. This section explains two segmentation method based on topological coordinate and types of writhe matrices.

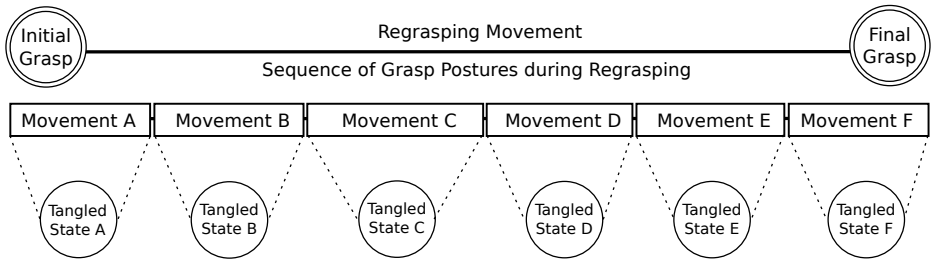


Figure 4.5: Regrasping movement as a transition of tangled states.

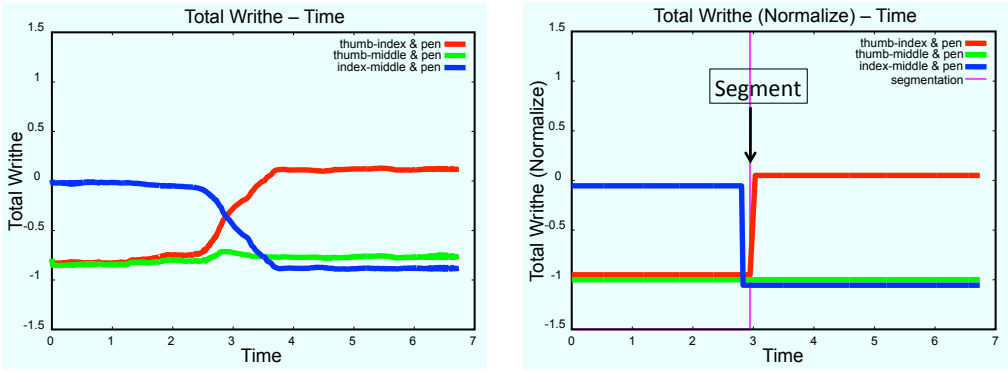
4.3.1 Method based on Topological Coordinate

Once a human hand and an object are represented as a combination of strands, grasping posture is simply a collection of tangles between these strands which can be described through a topological coordinates. In a similar manner, the regrasping movement can be perceived as a sequence of topological coordinates between the hand and the manipulated object over time. By observing a human hand during regrasping and analysing the tangle relation between the strands of a human hand and the object, it can be seen that there is a time period where tangle relationships of every pair of strands are not changed. In other words, if a *Tangled State* describes an unspecific-length time period where some specific attributes of topological coordinates of every concerned pair of strands remain, somewhat, constant, the regrasping movement can then be referred to as a transition of tangled states as shown in Figure 4.5.

A regrasping movement is segmented into many shorter segments, where each segment is referred to as a tangle state. Within a state a specific attribute of topological coordinates for related strand pairs remain consistent. In this subsection, segmentation methods based on each attribute of topological coordinates are described.

A human hand is represented with *alternative representation* explained in Section 3.2. In this method, only the tangle relation between three strands of the hand $\{S_{thumb-index}, S_{thumb-middle}, S_{index-middle}\}$ and a strand of the pen are considered. They are the strands that represent a connection between thumb, index and middle finger. By considering only the three fingers, the complexity of analysing a tangle relation between a human hand and the manipulated object is greatly reduced as well as the complexity of the proposed segmentation method.

Considering a pair of strands (S_h, S_r), S_h one of the three strands $\{S_{ti}, S_{tm}, S_{im}\}$ from a strand structure of human hand and S_r is a strand representing a pen, writhe value of this pair of strands is described by $w_{S_h S_r} = GLI(S_h, S_r)$. Since it is almost impossible for humans to use only two fingers to wrap around any



(a) Writhe value.

(b) Normalised writhe value and segmentation point in the movement.

Figure 4.6: Sample graphs of writhe and its normalised of three pairs of strands: red= thumb-index and pen, green= thumb-middle and pen, blue= index-middle and pen

object more than one round, the normalised writhe $\bar{w}_{S_h S_r}$ is defined as :

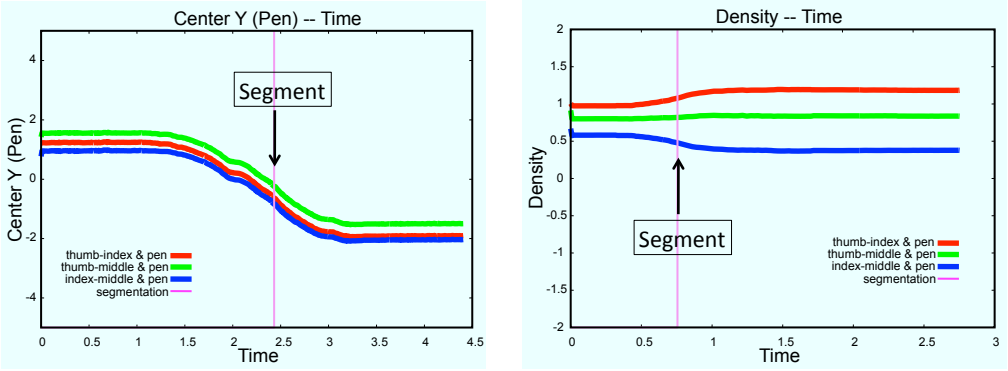
$$\bar{w}_{S_h S_r} = \begin{cases} 1 & w_{S_h S_r} \geq \delta \\ 0 & -\delta \leq w_{S_h S_r} < \delta \\ -1 & w_{S_h S_r} < -\delta \end{cases} , \tag{4.3}$$

where $0 < \delta < 1$ is a threshold value to separate whether there is a tangle between the pair and the plus/minus sign indicates the direction of the tangle.

Considering the three pairs of strands concerned, $\{S_{ti}|S_{tm}|S_{im}\}$ pairs with S_r, their writhe over time during a sample regrasping movement is shown in Fig. 4.6(a) and their normalised writhe over time in Fig. 4.6(b). In the later figure, the regrasping movement is segmented at the point where there are *critical* changes in two graphs from the total of three. The critical change in each graph is recognised by the local minimum/maximum in its first derivative. It can be seen that within the same segments every grasp posture has similar writhe relation.

There is also an analogy between the segmentation points based on writhe and the hand movement. These segmentation points can be interpreted as a movement where one of the fingers, either thumb, index, or middle finger, is purposely moving as to change its grasping position around the strand of the object. As a result, there are changes in the tangle relation of two strand pairs. One strand pair is changing from no tangle to one tangle around each other and the opposite is true for another strand pair.

Depending on the type of regrasping movement, the location on the pen where fingers are grasping can be greatly altered during the regrasping movement. For these kinds of movement, segmentation can be done based on an attribute center of topological coordinates. Center is a two dimensional vector. Each



(a) Center value and segmentation point in the movement.

(b) Density value and segmentation point in the movement.

Figure 4.7: Sample graphs of center and density of three pairs of strands: red= thumb-index and pen, green= thumb-middle and pen, blue= index-middle and pen.

of its elements illustrates an average location of tangle on one strand made by another strand. The element that describes the location of the tangle on the pen is analysed in the regrasping movement. Unlike the value of the tangle which describes by writhe, the location of the tangled area is a continuous attribute and cannot be broken down into finite feasible values. Therefore, the segmentation is performed at the points where one or more fingers are moving along or about the strand of the pen. They can be recognised by considering a high alteration of the center value in two or more graphs as shown in Fig. 4.7(a).

In a similar manner, an attribute density of topological coordinates can be used to segment a particular kind of regrasping movement. A segmentation based on density is used when a tangle (formed by fingers and the pen) changes shape over time. Density describes an angle between two axises in the writhe matrix, so its domain ranges from $[0, \pi)$. In the case of regrasping a rigid body object, strands of the object cannot wrap around strands of the manipulating hand. Therefore, the range of density is limited to $[0, \frac{\pi}{2})$. However, this does not have a significant effect on the designed segmentation method. Similar to center, density is a continuous value, although it has a very limited range, but still cannot be broken down into finite states. The segmentation points are thus defined as the high alteration of density in two or more of the three graphs as shown in Fig. 4.7(b).

Once the movement is segmented, based on either of the three attributes, into many shorter segments, every grasp posture in each segment are considered to be in the same tangled state because of their similar characteristics in topology space. For each segment, a grasp posture in the middle of the segment is chosen as its representative grasp posture. As a consequence, the original movement is

now represented with a sequence of these representative grasp postures which represents its change in topological relation during regrasping.

4.3.2 Method based on Types of Writhe Matrices

When a writhe matrix represents a tangle relation between a finger and a pen, it is possible to be characterised into either peak-wm or span-wm. Peak-wm represents a situation where a finger is in contact with a pen, and span-wm otherwise as mentioned earlier in Section 4.2.2. Regrasping movement can be thought of as a sequence of these writhe matrices, repeatedly changing from one type to the other. The proposed method segments a regrasping movement into many shorter movements, where all writhe matrices in every shorter movements are the same type. The segmentation is done independently for a sequence of writhe matrices of each finger. In other word, the proposed method can be perceived as a method to segment a regrasping movement based on the changes of contact relation between hand and object. It does not use the tactile sensor to detect this, but instead use tangle relation between a finger and a pen.

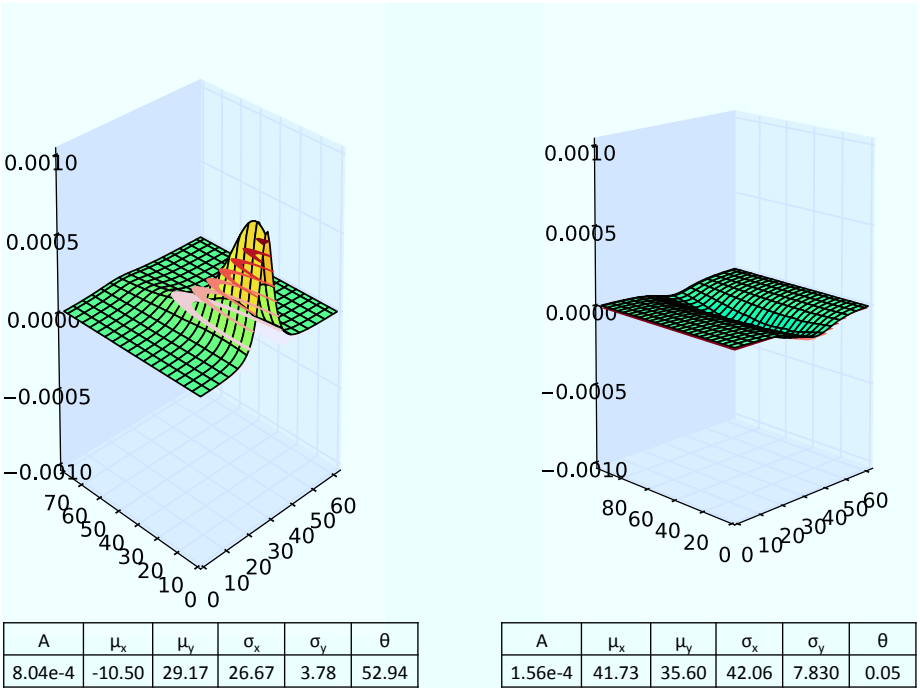
In short, the following process is used to segment a regrasping movement. After all writhe matrices of grasping postures in the sequence are calculated, each of them is fitted to a non-singular Bivariate Gaussian distribution. Parameters obtained from the fitting is use to characterise one writhe matrix to the others. The segmentation method makes use of these parameters to decide when the changes in type of writhe matrices occurs.

4.3.2.1 Fitting Writhe Matrix

Consider two strands S_1 and S_2, each strand is represented by line segments of n_1 and n_2 segments respectively. As explained in Section 3.1, a writhe matrix (T) is a $n_1 \times n_2$ matrix whose element T_{ij} is the writhe between segment i of S_1 and segment j of S_2. The distribution of these T_{ij}s is modelled as a non-singular Bivariate Gaussian function. Parameters of a particular writhe matrix is defined as parameters of the function. In order to discover these parameters, a writhe matrix is fitted to a Bivariate Gaussian function using a least square method.

A non-singular Bivariate Gaussian function is described as the following [Hagen & Dereniak 2008]:

$$f(x,y) = A\mathrm{e}^{-(a(x-\mu_x)^2+2b(x-\mu_x)(y-{}^m u_y)+c(y-\mu_y)^2)},$$



A	μ_x	μ_y	σ_x	σ_y	θ
8.04e-4	-10.50	29.17	26.67	3.78	52.94

A	μ_x	μ_y	σ_x	σ_y	θ
1.56e-4	41.73	35.60	42.06	7.830	0.05

Figure 4.8: Result of fitting bivariate Gaussian distribution to writhe matrices shown in Fig. 4.3. Resulting parameters are shown under each graph, and the layer in brown, are the Gaussian distribution reconstructed from the resulting parameters.

where

$$
\begin{aligned}
a &= \frac{\cos^2\theta}{2\sigma_x^2} + \frac{\sin^2\theta}{2\sigma_y^2} \\
b &= -\frac{\sin 2\theta}{4\sigma_x^2} + \frac{\sin 2\theta}{4\sigma_y^2} \\
c &= \frac{\sin^2\theta}{2\sigma_x^2} + \frac{\cos^2\theta}{2\sigma_y^2}.
\end{aligned} \tag{4.4}
$$

Equation (4.4) have six parameters, $(A, \mu_x, \mu_y, \sigma_x, \sigma_y, \theta)$. The explanation of all parameters are summarized in Table 4.1.

A least-square method is used to find the most appropriate parameters to fit a particular writhe matrix.A Levenberg-Marquardt algorithm is used to find the optimized parameters. For fitting writhe matrix $T_{n_1 \times n_2}$, the objective function for the optimization is given in Equation (4.5) as

Parameter	Description
A	amplitude of the peak
μ_x	mean in x direction
μ_y	mean in y direction
σ_x	standard deviation in x direction
σ_y	standard deviation in y direction
θ	rotation angle (clockwise)

Table 4.1: Description of parameters for Bivariate Gaussian function.

$$\underset{(A,\mu_x,\mu_y,\sigma_x,\sigma_y,\theta)}{\text{minimize}} \quad \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\left(T_{i,j} - f(i,j)\right)^2, \tag{4.5}$$

where $f(i,j)$ is a Bivariate Gaussian function defined in Equation (4.4). Notice from Equation (4.5) that the objective function allows the writhe matrix to be partially fitted to the Gaussian function.

Fingers of human hand are represented by strands as shown in Figure 3.5(a). Intuitive method of dividing a strand into line segments is by having a line segment connecting between two consecutive joints. However, all line segments that represent same strand would have different strand length. This causes its corresponding writhe matrix to be very different from a Bivariate Gaussian function. The issue is solved by further divided all line segments into smaller line segments until they have the same (similar) length. This can be performed without having any effect on the total writhe as GLI satisfies a distributive property over the operation of concatenating two strands [Ho & Komura 2009b]. The algorithm used to normalize length of line segments in the strand is similar to an algorithm used to approximate a greatest common divisor, and is given in Algorithm 1.

Granularity of how the strand is divided into line segments is another issue. The more the number of line segments is divided, the more the number of elements in writhe matrix to be fitted. If strand S is divided into a different number of line segments, it may result in a different outcome of fitting parameters. This issue is solved by applying a scale space technique [Witkin 1983]. Strands of a finger and a pen is divided into a shorter normalized line segments using Algorithm 1 with a range of desired length. Writhe matrix is calculated for each desired length, and is fitted to the Gaussian function. The resulting parameters are compared, and the final desired length is chosen to be the shortest one that the resulting parameters are started to be constant.

Figure 4.9 shows a graph between a number of line segments used to represent a strand of a (middle) finger and a resulting parameter (μ_x) from a Bivariate Gaussian fitting. It is noticeable that when a number of line segments is increased

Algorithm 1: NLength Normalize length of line segments in a strand

Input: $S_a = [s_1, \ldots, s_m]$; List of line segments representing strand S
Input: l; Desired length of line segments, where $\forall s_k \in S_a, l <= \|s_k\|$
Output: $S_b = [s_1, \ldots, s_n]$; List of normalized line segments representing strand S, where $\forall s_k \in S_b, l \approx \|s_k\|$

$S_b \leftarrow [\ ]$
% For every line segments in the initial set,
% divide it into a smaller line segments.
for $s_k \in S_a$ **do**
    % Explanation of each sub-sequential line :
    % - Find # of smaller line segments to be divided.
    % - Find new length for each smaller line segment.
    % - Divide s_k into num segments with length l_{new}.
    % - Combine result to output list.
    $num \leftarrow floor\left(\|s_k\|/l\right)$
    $l_{new} \leftarrow l/num$
    $[t_1, \ldots, t_{num}] \leftarrow divide(s_k, l_{new})$
    $S_b \leftarrow append(S_b, [t_1, \ldots, t_{num}])$
return S_b

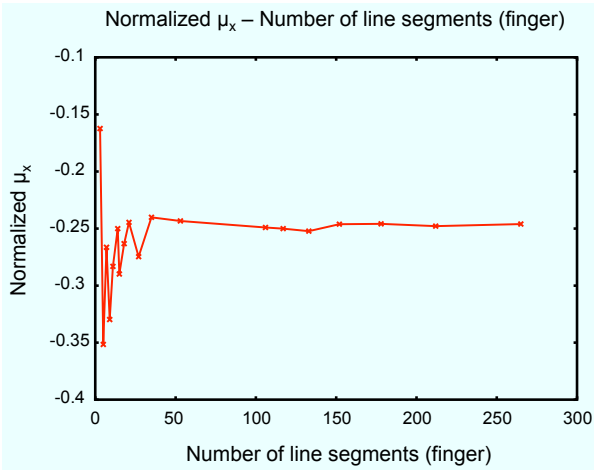


Figure 4.9: Considering a grasping posture in Figure 3.6(a), writhe matrices of a middle finger and a pen are calculated by varying a number of line segments that used to represent a strand of the finger. Each writhe matrix is then fitted to a Bivariate Gaussian function. Graph shows a relation between a number of line segments and a resulting parameter (μ_x) after normalized to the same scale ($\mu_x \in [-1, 1]$).

Type	$\sigma_{\mathbf{x}}$	$\lvert\mathbf{w}\rvert$
Peak-wm	low	high
Span-wm	high	low

Table 4.2: Relationship between two types of writhe matrices and their parameters.

to a certain value, the resulting parameter has become stabilised.

4.3.2.2 Segmenting Regrasping Movement

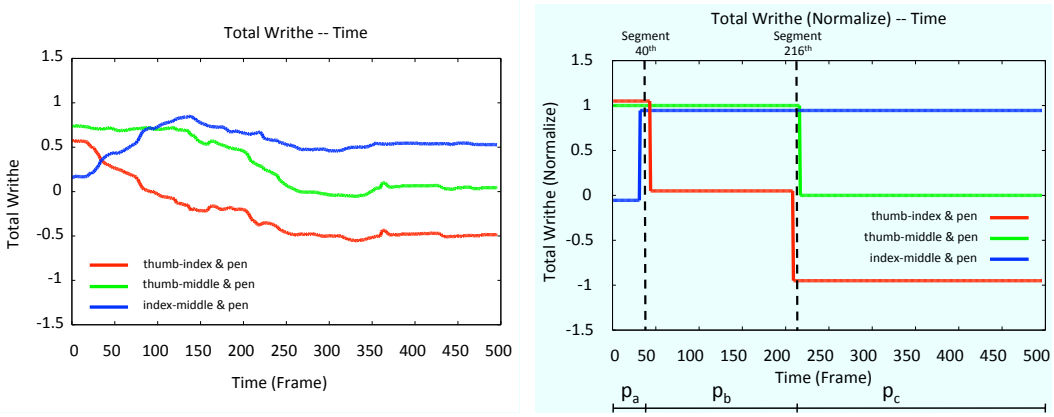
The segmentation is done so that all writhe matrices in every shorter movement are the same type (either peak-wm or span-wm). When considering writhe matrices in regrasping movement, the change in writhe matrix are continuous. In fact, there is no clear line to separate between peak-wm and span-wm, especially when a peak-wm is about to change to span-wm during detaching. To solve this issue, thresholding is used to create a clear separation for two types of writhe matrices.

Two parameters, a standard deviation of writhe matrix along a finger σ_x obtained the fitting and a total writhe w, are considered in this segmenting process because of the way writhe matrices are categorised into two types in the first place. In the case of peak-wm, when the finger are in contact with the object, the value of total writhe is usually higher than that of the span-wm. On the contrary, the value of σ_x of peak-wm is supposed to be lower than that of the span-wm. This is because when a finger is in contact with the object, the high values of writhe along the finger direction seems to be concentrated only in the vicinity of the contact location. The relationship of both types of writhe matrices and theirs parameters is summarized in Table 4.2.

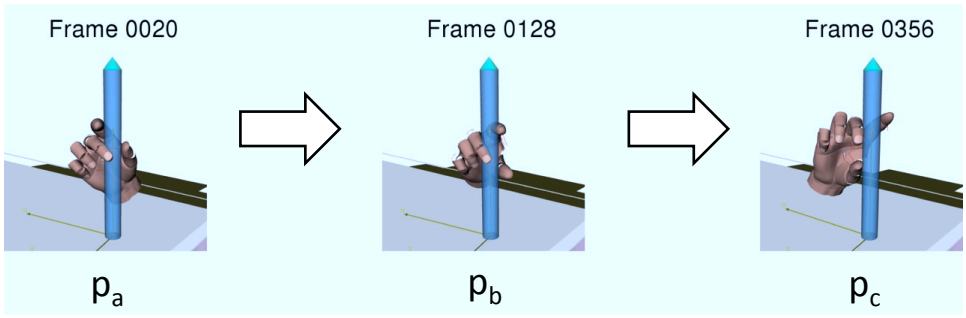
A combination of the two parameters $\sigma_x / \lvert w \rvert$ is used as a feature for a segmentation. Captured regrasping movement is turned into a sequence of writhe matrices. Hysteresis thresholding is applied on a temporal sequence of features. The higher threshold is related to span-wm, while the lower threshold is related to peak-wm.

4.4 Experimental Results

Two sequences of human regrasping movement are captured using a data acquisition system described in Appendix A. They are both represented an Interdigital Step movement as illustrated in Appendix C. Both analysis and segmentation methods are applied to both input sequences and the results are shown sequentially in the following subsections. The section is concluded with a comparison of the results.



(a) Graph of total writhe, its normalised and the segmented points. Three coloured lines represent a writhe (or a normalised writhe) between pairs of strands: red= thumb-index and pen, green= thumb-middle and pen, and blue= index-middle and pen. The movement is segmented at frame 40^{th} and 216^{th}.



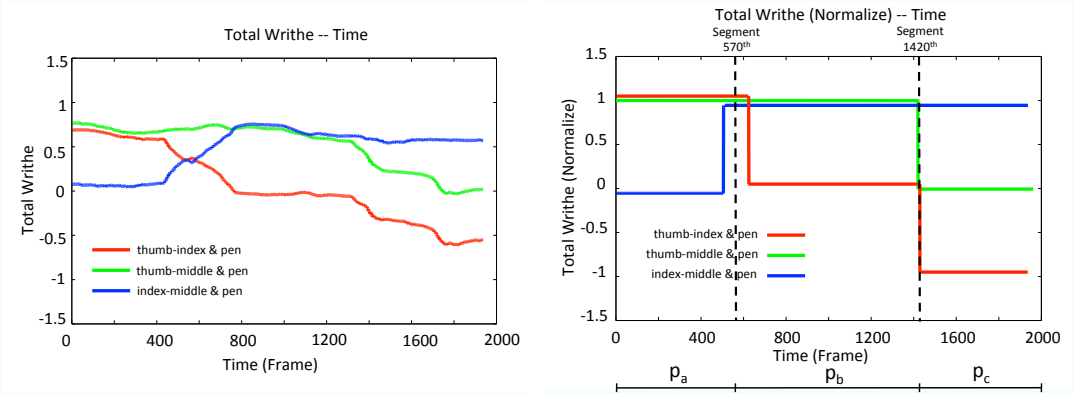
(b) Grasping postures, at the middle of each segment, represent each tangle states.

Figure 4.10: Topological analysis and segmentation of Interdigital Step – Sequence 1.

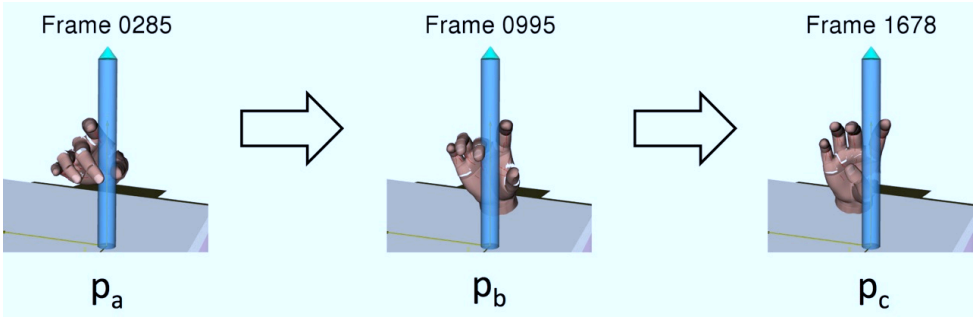
4.4.1 Results based on Topological Coordinate

Figure 4.10 shows an analysis and a segmentation result of Interdigital Step – Sequence 1. Graphs in each colour shows a writhe (or a normalised writhe) of a pair of strands, one from the hand and the other from the pen. A threshold value, $\delta = 0.3$, is used in Equation (4.3) to discretise decide whether there is a tangle between the pair. The movement is segmented at frame 40^{th} and 216^{th} based on the normalised writhe value. At the segmented frames, it can be seen that there are critical changes in two graphs from the total of three. The segmented frames divide the movement into three shorter segments, which respectively assign into tangle states p_a, p_b, p_c due to the similarity in their writhes of the three pairs of strands within the segments.

Similarly, Figure 4.11 shows an analysis and a segmentation result of Interdigital Step – Sequence 2. The threshold value, $\delta = 0.3$, is used to discretise the



(a) Graph of total writhe, its normalised and the segmented points. Three coloured lines represent a writhe (or a normalised writhe) between pairs of strands: red= thumb-index and pen, green= thumb-middle and pen, and blue= index-middle and pen. The movement is segmented at frame 570^{th} and 1420^{th}.



(b) Grasping postures, at the middle of each segment, represent each tangle states.

Figure 4.11: Topological analysis and segmentation of Interdigital Step – Sequence 2.

writhe value. The result shows a similar result with the two segmented frames, 570^{th} and 1420^{th}, and three tangle states.

Regarding the choice of the threshold value, δ, it is chosen to discretise whether there is a tangle (or a reversed tangle) between the pair of strands or not. It can seen from the graph that when there is a tangle between a particular pair of stands, the total writhe of the writhe matrix is between $0.5 - 0.8$ which is corresponded to what GLI is described. This mean that if the threshold is chosen to be a little more than zero or a little less than 0.5, the result would still be very similar.

The segmentation divides regrasping movement into a sequence of tangle states. Within each state, the normalised writhes ($\bar{w}$) of all strand pairs are same throughout. For instance, in Interdigital Step – Sequence 1, state p_a represents grasping postures that there is a tangle ($\bar{w} = 1$) between two pairs of strands $S_{ti} - S_r$ and $S_{tm} - S_r$; while there is no tangle ($\bar{w} = 0$) between a pair of

strands $S_{im} - S_r$. From this knowledge, the movement between states can also be interpreted as well, e.g. a movement from p_a to p_b represents a movement where an index finger moves from one side of the pen to the other, getting ride of a tangle between $S_{ti} - S_r$ and creating a tangle between $S_{im} - S_r$.

4.4.2 Results based on Types of Writhe Matrices

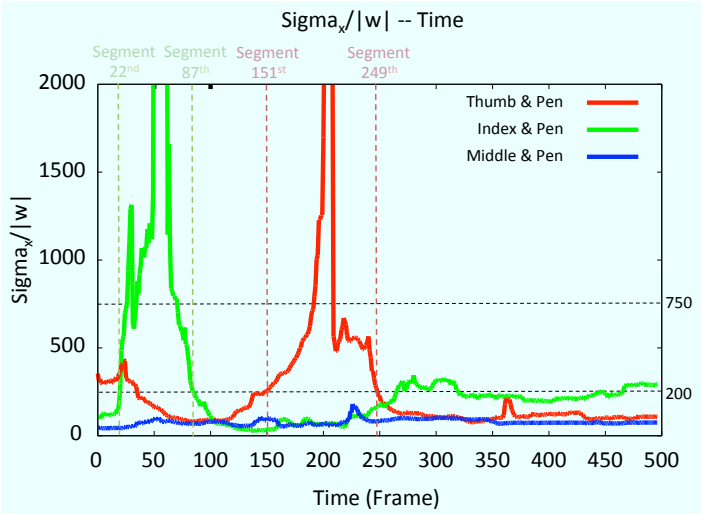
Figure 4.12 shows a graph of $\sigma_x / |w|$ of the sequence of writhe matrices between the three fingers and the pen and the segmentation result based on two types of writhe matrices of Interdigital Step – Sequence 1. All writhe matrices in the sequence are fitted with a Bivariate Gaussian function, as explained in Section 4.3.2.1, to obtain their σ_xs. Hysteresis thresholding is used to segment each coloured line independently to identify when there is a change in type of writhe matrix, either from peak-wm to span-wm or span-wm to peak-wm. A high threshold of 700 and a lower threshold of 200 is used. At the segmented frames, it indicates a moment when the finger starts to have or to lose a contact with the pen. The segmented frames for the red line are frame 151^{st} and 249^{th} as a changes of contact of the thumb, and for the green line are frame 22^{nd} and 87^{th} as a changes of contact of the index finger.

Similarly, Figure 4.13 shows a $\sigma_x / |w|$ graph and the segmentation result of Interdigitral Step – Sequence 2. Same thresholds are used for the segmentation. Similar results can be seen except the frame number of the segmented frames.

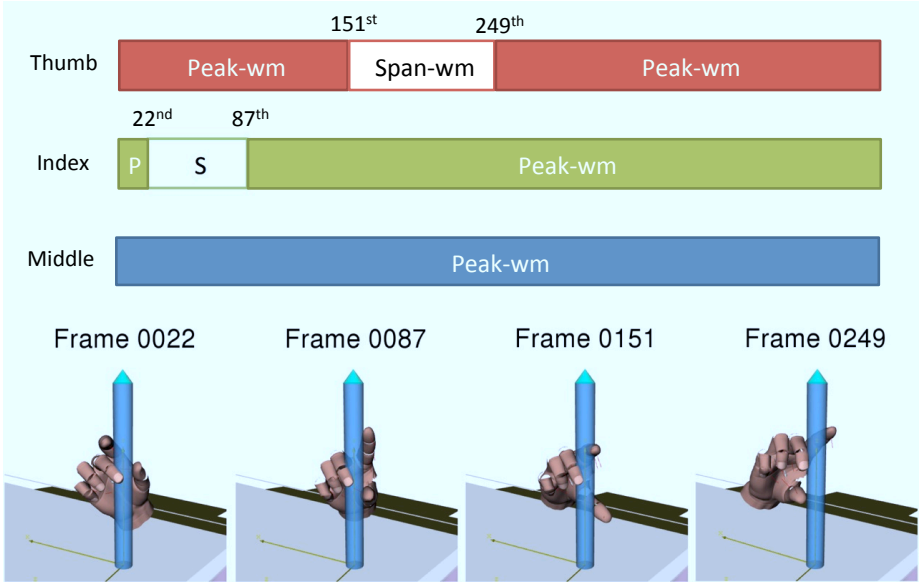
Regarding the choice of the threshold values, they are empirically decided. A lower threshold indicates a value when the finger is in contact, and the higher threshold to handle the case when there is noise from the data acquisition system. For instance, in Figure 4.12(a) it can be seen that in the beginning of the red line, the value spikes up due to the inaccuracy of the capturing system (the thumb should be in contact, but it isn't. This can see very clear in frame 35^{th} - 50^{th} in Appendix C.1.1). Therefore, the downside of the method is that these threshold values have to be carefully chosen, unless the accuracy of the data acquisition system is improved.

4.4.3 Comparison of Segmentation Results

It is noticeable that results from both methods are interpreted in totally different manners. Results based on topological coordinate consider a regrasping movement into a sequence of tangle states. Although the method could recognise the finger that move to change its orientation to the pen, it cannot identify when the movement started to occur. The grasping postures shown in both Figure 4.10(b) and Figure 4.11(b) are only chosen from the middle of the segments to show the characteristic of the corresponding tangle states. On the other hand, results based on differentiating types of writhe matrix segment a regrasping movement



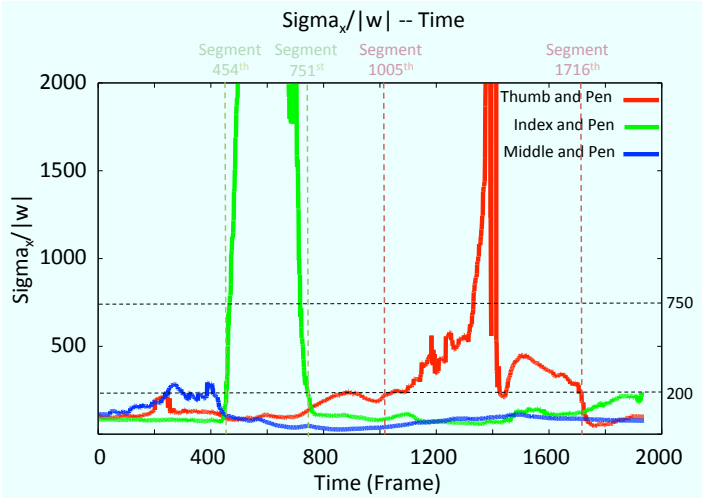
(a) Graph shows $\sigma_x / |w|$ of the sequence of writhe matrices of between three fingers and the pen: red = thumb and pen, green = index and pen, blue = middle and pen. The movement is segmented at frame 151st and 249th for the red line, and is segmented at frame 22nd and 87th for the green line.



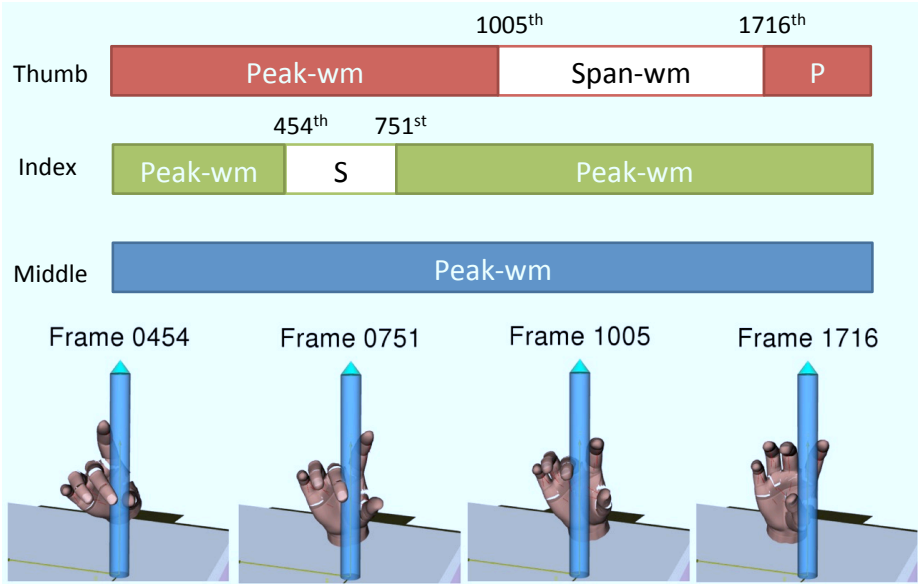
(b) Interpretation of the type of writhe matrices. The colours are corresponded to Figure 4.12(a). Grasping postures of the segmented frames are also illustrated, representing the movement where writhe matrices change their type.

Figure 4.12: Graph of a combination of features of writhe matrices and the segmentation of Interdigital Step – Sequence 1.

into a sequence of segments based on contact relations of the fingers. The method



(a) Graph shows $\sigma_x / |w|$ of the sequence of writhe matrices between three fingers and the pen: red = thumb and pen, green = index and pen, blue = middle and pen. The movement is segmented at frame 1005^{th} and 1716^{th} for the red line, and is segmented at frame 454^{th} and 751^{st} for the green line.



(b) Interpretation of the type of writhe matrices. The colours are corresponded to Figure 4.13(a). Grasping postures of the segmented frames are also illustrated, representing the movement where writhe matrices change their type.

Figure 4.13: Graph of a combination of features of writhe matrices and the segmentation of Interdigital Step – Sequence 2.

recognises whether at the particular moment the fingers have contact relations with the pen or not.

The information on contact relation is very crucial for mapping the movement onto the robotic hand. This can be seen from the literature presented in Section 4.1. In fact, the experiment on mapping the regrasping movement to the robotic hand using a topological coordination has been conducted [Vinayavekhin *et al.* 2011]. The results indicate that to imitate human regrasping movement in a physical world, an information on contact locations are very crucial. Topological coordinate might allow movements of finger to be easily interpolated, but to retain the stability of the object, contact relation must be considered.

4.5 Summary of Chapter

Human regrasping movement is analysed and segmented with features in tangle topology. Two segmented methods are described. One is based on an attribute of topological coordinate, while the other is based on a classification of writhe matrices into two types: *peak-wm* and *span-wm*. Peak-wm represents a writhe matrix when a corresponding finger is in contact with the object, and span-wm otherwise. The segmented results from both methods can be interpreted differently. The former represents human regrasping movement by a sequence of tangle states, where some attribute of topological coordinate is constant within the state. The latter represents human regrasping movement as changes in type of writhe matrices, which can referred to as changes of contact states of fingers during regrasping.

CHAPTER

FIVE

Recognition of Regrasping Movements

Contents

In this chapter, a method to recognise human regrasping movement is presented. Robot recognises the regrasping movement to understand what is important and learn how to imitate it. This is done based on a pre-defined model, referred to as a task model. The chapter starts by defining a task model for a RPO system. Then a method to recognise the task model from human grasping movement is described. The experimental results of the recognition method is given. They are intermediate results that will be later used to map the movement onto the robotic hand.

5.1 Task Model

A task model is very important element of the RPO system. It consists of two main components: movement primitives and skill parameters. Movement

primitives define what the robot suppose to do in order to imitate human, and skill parameters tell the robot how to do that. The definition of the task model is based on the classification that classify writhe matrices into two types: peak-wm and span-wm.

According to what had been described in Section 2.1.1, in order to define a set of movement primitives for a specific task or movement, it needed to be characterised and classify into recognisable states. Then these states can be classified or grouped into one type of movement primitive. Therefore, the section starts by describing a grasp taxonomy used to classify human grasping posture. Then the definition of movement primitive and skill parameters are followed respectively. Finally, it explains how the regrasping movement can be represented once a task model is defined.

5.1.1 Classification of Grasping Postures

Regrasping movement is captured as a sequence of grasping postures by a data acquisition system. Before a set of movement primitives, which is a main part of a task model, can be defined, there must be a method to characterise this sequence into states. To satisfy this, a taxonomy of grasping postures is required.

There are quite a few studies in the literature on a classification of grasping postures. Napier divided grasping postures broadly into two categories: power and precision grasp [Napier 1956]. The idea was based on the distinction of their anatomy and functionality. However, some prehensile movements could also be considered in both categories. Cutkosky adopted this idea and extended it into a detailed taxonomy of human grasping postures [Cutkosky 1989]. The taxonomy is a hierarchical tree. Different grasping postures in the hierarchy are differentiated by their functionalities (task requirements) and a geometry of the manipulated object.

Kamakura *et al.* also proposed a classification of grasping postures [Kamakura *et al.* 1980]. However, their categories were derived from differences in contact areas, both on human hand and the object. They suggested that grasping postures should be able to classify regardless of the knowledge of activities (task requirements). Kang and Ikeuchi followed similar ideas and proposed another grasp taxonomy [Kang & Ikeuchi 1993]. Their classification utilized an information about the hand configuration and the object shape, without any knowledge about task requirements. They classified grasping postures based on a *contact web*, a 3D graphical structure of contact location on the hand.

In RPO system, the taxonomy will be used to characterise grasping postures in a regrasping movement. These grasping postures could be any posture that does not possess any meaning. They might be just any intermediate posture that leads to the final posture in the movement. This is why those taxonomies that illustrate grasping postures based on their task require-

ments [Napier 1956, Cutkosky 1989] are not suitable. Furthermore, these intermediate grasping postures could have some fingers that are not in contact with the object, but moving to change its contact. Imitating regrasping movement is also about representing these postures. This mean that those taxonomies that based solely on contact location [Kamakura *et al.* 1980, Kang & Ikeuchi 1993] are not applicable as well.

A taxonomy used in RPO system is described in the following subsections. The taxonomy is based on a geometric property that describes a relation between two strands, a GLI. The taxonomy is proposed based on the changes of type of writhe matrices between fingers and object.

5.1.1.1 A Taxonomy of Grasping Posture

The taxonomy for RPO system is developed based on different types of writhe matrices of the hand. To reduce the complexity, the taxonomy will consider grasping postures that relate to only three fingers: thumb, index finger, and middle finger. At the particular moment, writhe matrix of each finger could be in one of the following four states: peak-wm(sign+), peak-wm(sign-), span-wm(sign+), and span-wm(sign-). This mean that, in general, for three finger case, the taxonomy would possess $4^3 = 64$ different states of grasping posture. However, when all other constraints applied, the taxonomy ends up having only 18 feasible states.

Before describing the taxonomy, a notation is given here for a concise descriptive purpose.

- F_T^S : represents writhe matrix of finger F, with a type T, having a sign S.
  - $F \in \{t, i, m\}$: Only thumb, index finger, and middle finger are considered.
  - $T \in \{p, s\}$: Type of writhe matrix can only be either peak-wm or span-wm.
  - $S \in \{+, -\}$: Sign of writhe matrix can only be either plus or minus.
- $t_p^+ i_p^+ m_p^- = m_p^- t_p^+ i_p^+$: Order of fingers makes no difference, but tim will be used throughout.

The taxonomy is derived as the followings:

1. Consider when the direction of the pen is fixed, sign of the writhe matrix indicates the side of the corresponding finger regarding to the pen. Since there are only three fingers involved, two of them must be on the opposite side in order to hold the pen. In other word, all three writhe matrices cannot have the same sign, regardless of their types. This makes all the possibilities to be $2^3 - 2 = 6$ cases, as listed below.

- $t^+i^+m^-$
- $t^+i^-m^+$
- $t^-i^+m^+$
- $t^-i^-m^+$
- $t^-i^+m^-$
- $t^+i^-m^-$

2. Among three writhe matrices, at least two must have the peak-wm type. Furthermore, those particular two must have the opposite sign. In other word, at least two fingers must be in contact and holding the pen. Therefore, there are three possibilities for each one of six cases listed previously, e.g.

   - for $t^+i^+m^-$, all possible states are $t_p^+i_p^+m_p^-$, $t_s^+i_p^+m_p^-$, and $t_p^+i_s^+m_p^-$.

Combining these two constraints, the taxonomy ends up having a totally of $6\times3 = 18$ possible states, or 18 categories of grasping posture used to manipulate the pen. A more perceptual understandable taxonomy is illustrated in Figure 5.1.

It can be seen that by comparing with other taxonomies explained early in the section, the proposed taxonomy described more about fingers that are not in contact with the object, i.e. the finger that was corresponded to span-wm. This is essential in a RPO system because the finger, that is not in contact, might be moving as a part the original regrasping movement and its role might need to be imitated. The concern about movement in grasp taxonomy can be seen in a recent proposal [Bullock *et al.* 2012]. Bullock *et al.* proposed a taxonomy that covered a movement of grasping postures. Their taxonomy is quite general, and can deal with any kind of object. Although RPO system is considered grasping postures in different context, the proposed taxonomy in a RPO system could be considered as a *Sub-Classification* of their taxonomy.

Another interesting of aspect of the proposed taxonomy is that it treats each finger as an individual entity. This is resembled to the concept of *Virtual Fingers* [Arbib *et al.* 1985]. Virtual finger represents a finger or a group of finger or a hand surfaces that work together as a unit and apply same direction of force or torque at the object. Currently, in the proposed taxonomy only one finger is grouped as an entity. However, an idea of applying virtual finger could be useful, if more fingers are to be considered into the taxonomy.

5.1.2 Movement Primitives

Based on the taxonomy, grasping postures are classified into 18 states as described in Section 5.1.1.1. In theory, there should be a totally of $\binom{18}{2} = 153$ transitions between every pair of states. However, when all constraints are taken into consideration (e.g. at least two fingers have to be in contact and hold the pen at all moment), there are only 18 feasible transitions as shown in Figure 5.1.

Each transition represents a movement that connects between two states of grasping postures. Eighteen transitions can be classified into three categories

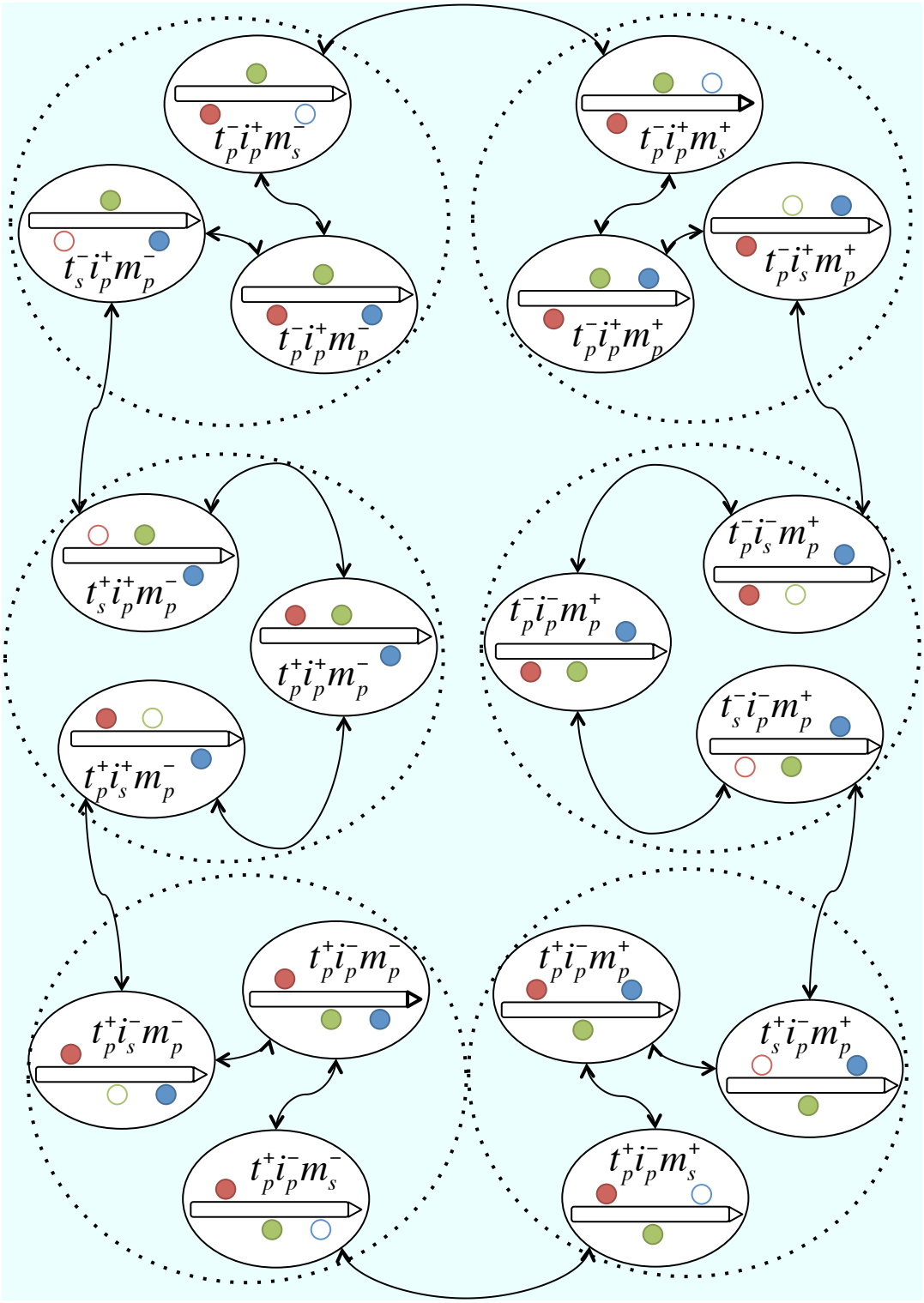


Figure 5.1: State-transition of RPO system. Taxonomy of grasping postures is constructed based on types of writhe matrices between fingers and a pen-like object. States (circled with ——) in a diagram represents all grasping postures in the taxonomy. Transitions connects between two states, representing a change of one grasping posture to the other. States are grouped (circled with - - -) based on signed of all writhe matrices.

based on the changes of writhe matrices during the movement. These categories of transition are referred to as movement primitives in the RPO system. Movement primitive is a main component of RPO task model. It is an abstract representation that connect between human demonstration and robot execution. Original regrasping movement is represented as a sequence of these movement primitives, which will then transfer to the robot to inform which movements should be imitated. Based on an observation, through these three movement primitives, the robot should be able to imitate any human regrasping movement of the pen-like object. In this section, all three movement primitives are described.

1. **Detaching** represents a movement where a finger moves away from an object. The writhe matrix changes from peak-wm to span-wm, within the same sign of writhe matrix.

2. **Attaching**, on the contrary, represents a movement where a finger moves toward an object in order to make a contact. The writhe matrix changes from span-wm to peak-wm, within the same sign of writhe matrix.

3. **Crossover** represents a movement where a finger changes from one side to another side of the corresponding strand that representing the object. The writhe matrix changes from span-wm to span-wm with the different sign of writhe matrix.

These movement primitives are defined based on changes in finger level. This is a reason why 18 transitions of movement can be grouped into three types of movement primitive. Two different transitions may have the same movement primitives for different fingers, e.g.

- $t_p^+ i_p^+ m_p^- \rightarrow t_s^+ i_p^+ m_p^-$ is detaching on thumb while $t_p^- i_p^- m_p^+ \rightarrow t_p^- i_s^- m_p^+$ is detaching on index finger,

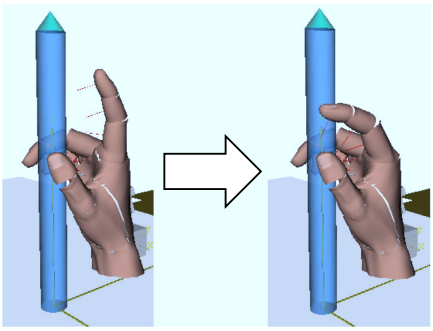
- $t_p^+ i_p^- m_s^- \rightarrow t_p^+ i_p^- m_p^-$ is attaching on middle finger while $t_s^- i_p^+ m_p^- \rightarrow t_p^- i_p^+ m_p^-$ is attaching on thumb,

- $t_p^- i_s^+ m_p^+ \rightarrow t_p^- i_s^- m_p^+$ is crossover on index finger while $t_p^+ i_p^- m_s^- \rightarrow t_p^+ i_p^- m_s^-$ is crossover on middle finger.

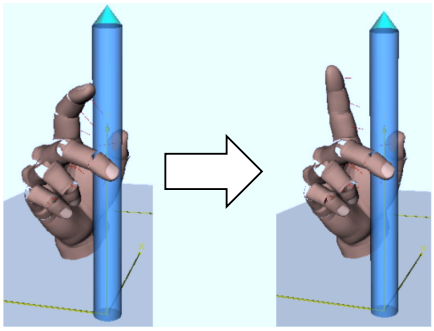
Figure 5.2 shows an example of all three movement primitives for the index finger when the manipulated object is a pen.

5.1.3 Skill Parameters

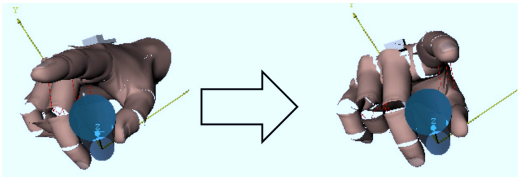
Beside movement primitives, skill parameters are also a main component for RPO task model. It tells the robot how to imitate each movement primitives. Each



(a) Detaching represents a movement when a writhe matrix between a (index) finger and a pen changes from peak-wm to span-wm, within the same sign of writhe.



(b) Attaching represents a movement when a writhe matrix between a (index) finger and a pen changes from span-wm to peak-wm, within the same sign of writhe.



(c) Attaching represents a movement when a writhe matrix between a (index) finger and a pen changes from span-wm with a plus/minus sign to span-wm with an opposite sign.

Figure 5.2: Grasping postures show three movement primitives of an index finger.

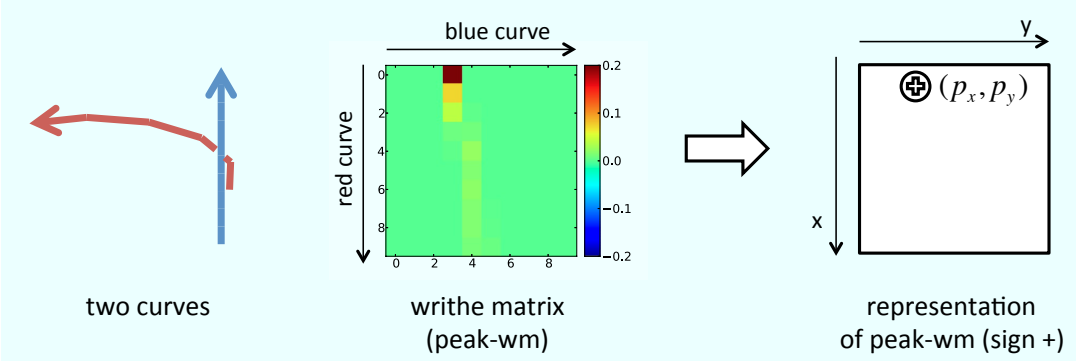
movement primitive need a different set of skill parameters to be executed. In this section, a set of skill parameters for each movement primitives is described.

Skill parameters for each movement primitive are mainly taken from writhe matrices of its initial and final states. The section will start by outlining a set of parameters used to describe each types of writhe matrix. Then by combining these informations and some additions, a set of skill parameters for each movement primitive are described.

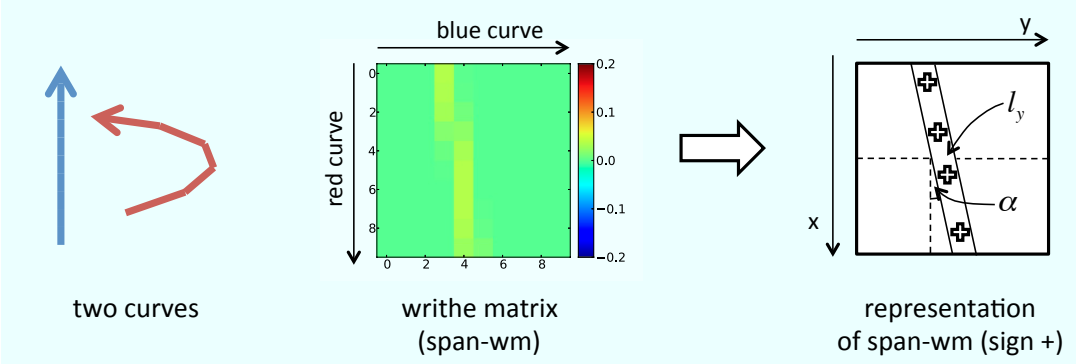
5.1.3.1 Parameters for Each Types of Writhe Matrix

Ho and Komura used writhe matrix to represent motions between characters [Ho & Komura 2009a]. All writhe matrices are treated the same in their system. A topological coordinate explained in Section 4.2.1 is used to describe all writhe matrices. However, this approach is not suitable for representing a regrasping movement as it needed a distinction of a contact relation. An evidence for this could be seen in our previous work [Vinayavekhin *et al.* 2011].

In RPO system, writhe matrix are characterised based on their appearances. Writhe matrices appear differently whether there is a contact relation between a



(a) Ideal case of peak-wm, where two curves are close to each other. It is noticeable that there is an area in writhe matrix that writhe value is high compared to the others. The region is referred to as peak, and parameterised by its location (p_x, p_y).



(b) Ideal case of span-wm, where two curves are, on average, apart from each other. Writhe is distributed across various area in the writhe matrix, and can be estimated as a line. It is referred to a span-line, and parameterised by its orientation (α, l_y).

Figure 5.3: Ideal case of two types of writhe matrix, and how they are parameterized.

finger and the object as explained in Section 4.2.2. Figure 5.3 shows two types of writhe matrix in an ideal situation where two strands are close or apart from each other. To describe each type of writhe matrix, a different set of parameter are used.

- **Peak-wm** is described by two attributes: writhe w and a location of a peak $\boldsymbol{p} = (p_x, p_y)$. The location of a peak indicates the area where contact occurs. p_x indicates location of contact on the finger, while p_y indicates location of contact on the object. For peak-wm T^p, its parameters will be referred to as $\mathcal{P}(T^p) = (w, p_x, p_y)$.

- **Span-wm** is described by two attributes: writhe w and a span-line $\boldsymbol{l} = (\alpha, l_y)$. The first part of the span-line, slope α, reflects the average orien-

Type	Parameter	Description
Peak-wm	$w \in \mathbb{R}$	total writhe
	$p_x \in [-1, 1]$	location of contact on a finger
	$p_y \in [-1, 1]$	location of contact on an object
Span-wm	$w \in \mathbb{R}$	total writhe
	$\alpha \in [0, \pi]$	average orientation between a finger and an object
	$l_y \in [-1, 1]$	average location on an object that is tangled

Table 5.1: Summary of parameters for each type of writhe matrix.

tation between the finger and the object. The other part l_y indicates the average location on the object that is tangled by the finger. A straight line is used to represent the orientation of the finger in this situation because when the finger is not in contact with the object, it makes little difference whether it is bent or not as shown in Figure 5.4(a). For span-wm T^s, its parameters will be referred to as $\mathcal{S}(T^s) = (w, \alpha, l_y)$.

Notice that the span-line does not contain directional information, which introduces two possible ambiguities as shown in Figure 5.4(b). However, the sign of writhe solves this ambiguity.

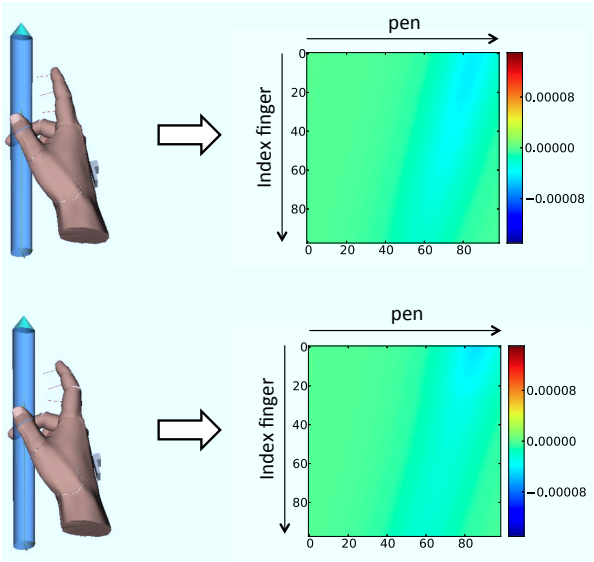
In addition, it is noticeable that some parameters of both types of writhe matrix are scaled to the range of $[-1, 1] \times [-1, 1]$. An explanation for this can be found in Section 6.2.1.1.

5.1.3.2 Parameters for Each Types of Movement Primitive

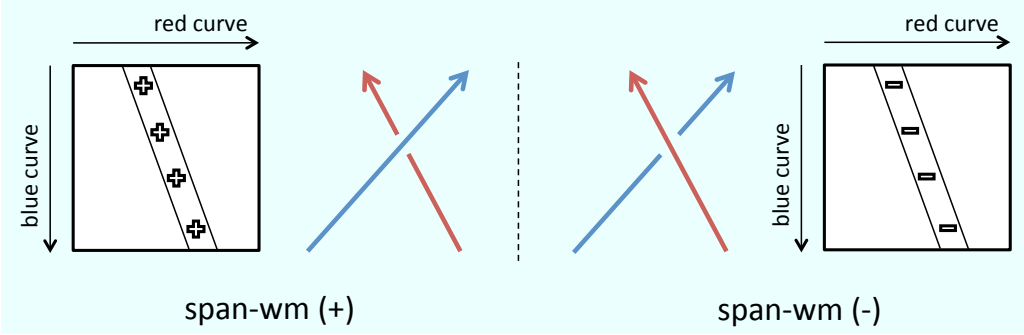
Most of skill parameters for each movement primitives are taken from its initial and final grasping postures. All the movement primitives have states of initial and final grasping postures as skill parameters. Each movement primitive also requires parameters that describes all three writhe matrices of both initial and final states. They are different depending on their type of movement primitive. These values determine the starting configurations of robot hand in mapping phase.

Another skill parameter for all types of movement primitive is an orientation of the pen. This information is necessary because writhe matrix does not contain this information, and the pen might be moving during a regrasping movement. Note that in a RPO system, the pen is assumed not to be moving within each movement primitives. However, the pen can be moved between two movement primitives. More details on this issue will be explained in Section 5.1.4.

Table 5.2 summarizes skill parameters for all movement primitives. As mentioned in Section 5.1.2, the major movement may occur in different fingers, even when referring to the same movement primitives. This finger is referred to as a *major moving finger*. It is a finger whose writhe matrix changes its type or its



(a) Two grasping postures are similar. The difference between them are the shape of an index finger which is not bent in the top case and bent in the bottom one. The index fingers results in similar span-lines of span-wms, which will not be differentiated when their span-wms are parameterised. The only difference between their parameters could only be a slight difference of a total writhe value.



(b) Orientation of two pairs of curves are the same, except the direction of the curves. They result in two similar span-wms. However, their sign of writhe are different which eliminate an ambiguity issue.

Figure 5.4: More explanations for a parameterisation of span-wm.

sign to the opposite one between the movement primitive. On the other hand, a *minor moving finger* is a finger that may move slightly, but its writhe matrix is kept as peak-wm with the same sign throughout the movement primitive. For instance,

Primitive	Parameter	Description
All	s_i	initial states (e.g. $t_p^+ i_p^+ m_p^-$)
(common-	s_f	final states (e.g. $t_s^+ i_p^+ m_p^-$)
parameters)	$\mathcal{P}_{i_1}$	params. of initial peak-wm of 1st minor finger
	$\mathcal{P}_{f_1}$	params. of final peak-wm of 1st minor finger
	$\mathcal{P}_{i_2}$	params. of initial peak-wm of 2nd minor finger
	$\mathcal{P}_{f_2}$	params. of final peak-wm of 2nd minor finger
	$\mathbf{v}_b \in \mathbb{H}$	orientation of the pen
Detaching	$\mathcal{P}_{i_0}$	params. of initial peak-wm of major finger
	$\mathcal{S}_{f_0}$	params. of final span-wm of major finger
Attaching	$\mathcal{S}_{i_0}$	params. of initial span-wm of major finger
	$\mathcal{P}_{f_0}$	params. of final peak-wm of major finger
Crossover	$\mathcal{S}_{i_0}$	params. of initial span-wm of major finger
	$\mathcal{S}_{f_0}$	params. of final span-wm of major finger

Table 5.2: Summary of skill parameters for each type of movement primitives.

- $t_p^+ i_p^+ m_p^- \rightarrow t_s^+ i_p^+ m_p^-$ is considered as detaching, and the major moving finger is thumb. The less are minor moving fingers.
- $t_p^- i_p^- m_p^+ \rightarrow t_p^- i_s^- m_p^+$ is also considered as detaching. However, its major moving finger is index finger, while the less are minor moving fingers.

5.1.4 Regrasping Movement as a Sequence of Movement Primitives

A regrasping movement of human hand is represented by a sequence of movement primitives. Each movement primitive connects between two states, an initial state and a final state. This mean that for two consecutive movement primitives, a final state of the former is the same as an initial state of the latter. However, this cannot guarantee that their corresponding grasping postures are exactly the same. This applies to all states in Figure 5.1 that have three peak-wm, i.e. all states with $t_p i_p m_p$. The method to connect the two grasping postures which are in these same states is discussed in Section 6.3.

Regarding a movement of the pen during regrasping, this information is also captured during observation. It can also be observed from human movement that when the pen movement occurs, a grasping posture do not change its state. In addition, the pen movement usually occurs in the states that have three peak-wm, i.e. all states with $t_p i_p m_p$. There is only a minority of pen movement occurs during any movement primitives. Thus, the conclusion regarding a movement of the pen in task model of the RPO system are the following:

- The pen is considered to be immobilised during each movement primitive is performed.
- The pen only moves when three fingers are in contact with the pen. In other words, writhe matrices between each finger and the pen are peak-wm. During the pen movement, contact location of some finger (thus some parameters of writhe matrices) and the grasping posture may change, but its state does not change.

5.2 Recognition of Task Model

A method to recognise a task model is presented in this section. A captured regrasping movement is recognised against the task model, in order to be represented as a sequence of movement primitives and skill parameters as described in Section 5.1.4. A recognition of task model is divided into two parts: a recognition of movement primitives from human regrasping movement, and an extraction of skill parameters for each movement primitive.

5.2.1 Recognition of Movement Primitives

A data acquisition system captures regrasping movement into a time series of grasping postures. This sequence of grasping postures is made up of various shorter sequences. Some of which are the movement that changes a state of grasping posture from one to another according to taxonomy explained in Section 5.1.1.1. These short movements, according to the definition in Section 5.1.2, are in fact the movement primitives.

To recognise the movement primitives in the regrasping movement, the captured regrasping movement is first segmented into many shorter movement using method described in Section 4.3.2. Once a sign and type of all writhe matrices of all three fingers are known, states of their corresponding grasping posture can be easily identified using taxonomy explained in Section 5.1.1.1. Finally, movement primitive can be identified by the change of these states as well.

5.2.2 Extracting Skill Parameters

After movement primitives are recognised, the human regrasping movement is represented semantically by a sequence of movement primitives. Skill parameters for each movement primitives, explained in Section 5.1.3.2, is extracted in order to be used during reproduction in a robotic hand.

In Section 5.2.1, movement primitives are recognised from the change of one state to another. However, a state is representing a range of grasping postures. In other word, many consecutive grasping postures are binded to a same states.

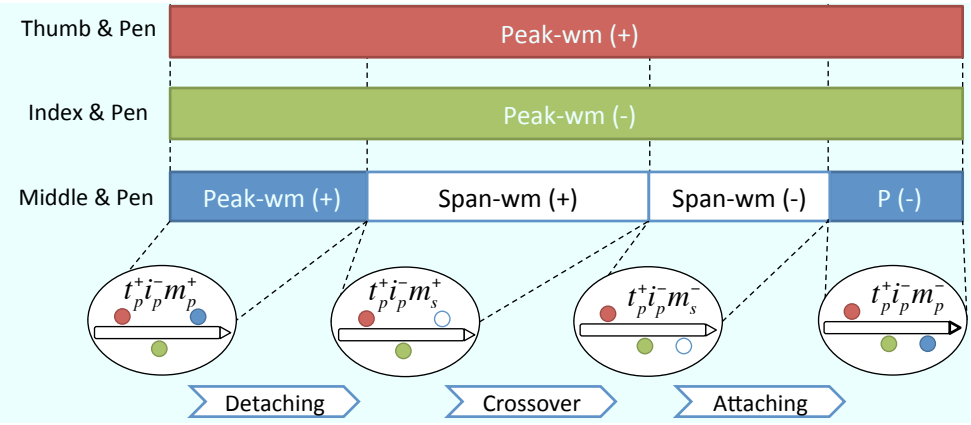


Figure 5.5: Example of identifying state of grasping posture from type of writhe matrix. Note that blue edges of movement primitives indicates that they belong to the movement of the middle finger.

The issue is that most of the skill parameters are extracted from initial state and final state, but not all parameters are not constant across the same state. In fact, only two parameters are constant across the same state which are parameters that indicate type of state themselves (referred to as s_i and s_j in Table 5.2). Therefore, before skill parameters of each movement primitive can be extracted, the first thing to do is to specify the range of the grasping postures that the primitive is spreading across. Then other skill parameters can then be extracted from the first and last grasping postures of the range.

5.2.2.1 Specifying a Range of Movement Primitive

All movement primitives always occur and are detected when a writhe matrix of one finger is span-wm. In other word, every time when there is a segment of span-wm occurs, it indicates that there is at least one movement primitive within that segment.

For all span-wm segments, there are four possible cases, as shown in Figure 5.6.

case 1 If the span-wm segment is in between two peak-wm segments and the sign of writhe changes to the opposite one, there are three movement primitives occurred in the segment which are *detaching* → *crossover* → *attaching*.

case 2 If the span-wm segment is connected to only one peak-wm segment and the sign of writhe changes to the opposite one, there are two possibilities in this case. When the segment is at the beginning of the hand movement, there is an occurrence of a sequence of *crossover* → *attaching* movement primitives. On the other hand, when the segments is at the end of the hand

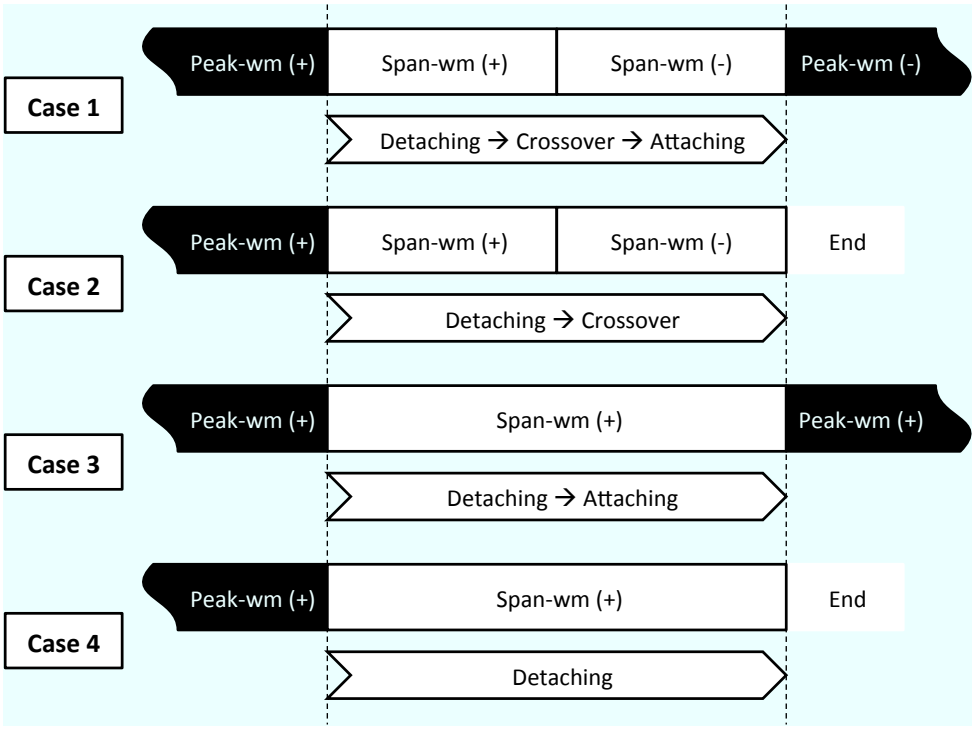


Figure 5.6: Four possible cases of movement primitives recognised in a span-wm segment.

movement, thus connected to peak-wm at the beginning of the segment, there is a sequence of *detaching* → *crossover* movement primitives occurred.

case 3 If the span-wm segment is in between two peak-wm segments and the sign of writhe value does not change, there is an occurrence of a sequence of *detaching* → *attaching* movement primitives.

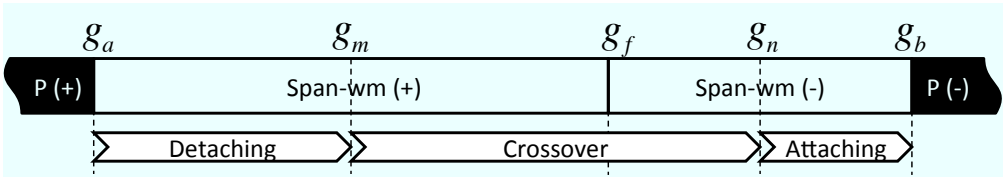
case 4 If the span-wm segment is connected to only one peak-wm segment and the sign of writhe value does not change, there is one movement occurred in the segment. When the segment is at the beginning of the hand movement, it is a *attaching* primitive. On the other hand, if the segment is at the end of the hand movement, it is a *detaching* primitive.

When a span-wm segment contains more than one movement primitive, the boundary of each movement primitive is needed to be specified. The ways the boundary is chosen are different depending whether there is a *crossover* primitive involved. When there is a *crossover* primitive in between a span-wm segment (in **case 1** and **case 2**), the span-wm segment is first divided into two shorter segments at the frame where the sign of its writhe value flips. Then, the first and

last grasping posture of *crossover* are taken from the middle frame of the two smaller segments. On the other hand, when there is no *crossover* primitive (thus no sign change in span-wm segment), the middle frame of span-wm segment is chosen as a segmented frame if required. Notice that when the span-wm segment is segmented, the middle frame is used in this case, as opposed to the ideal case where the frame that the finger is farthest from the object would be chosen.

Assuming g_a and g_b are the first and last frames of the span-wm segment, an explanation for specifying a range of movement primitives in all four cases of span-wm segment are given as the following.

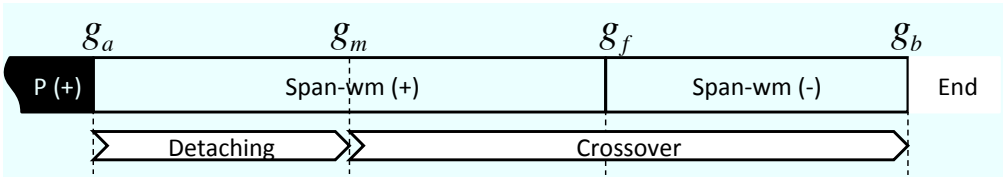
case 1 There are three movement primitives in this segment, *detaching* → *crossover* → *attaching*. Assuming g_f is a frame where a sign of writhe is equal to zero (flips to opposite sign), g_m is a mid frame between g_a and g_f, and g_n is a mid frame between g_f and g_b, a range of each movement primitive are given in Table 5.3.



Primitive	First	Last
detaching	$g_a - 1$	$g_m - 1$
crossover	g_m	g_n
attaching	$g_n + 1$	$g_b + 1$

Figure 5.7 & Table 5.3: A range of movement primitives in **case 1** of span-wm segment.

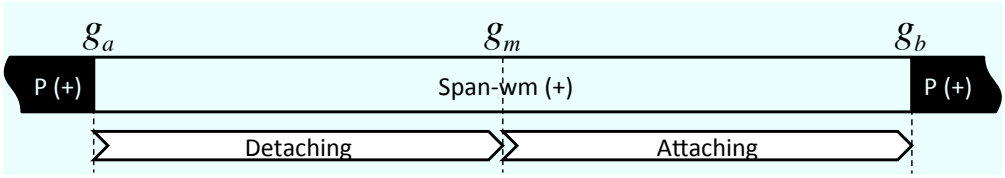
case 2 Considering the segment shown in Figure 5.6, there are two movement primitives in the segment, *detaching* → *crossover*. Assuming g_f is a frame where a sign of writhe is equal to zero (flips to opposite sign) and g_m is a mid frame between g_a and g_f, a range of each movement primitive are given in Table 5.4.



Primitive	First	Last
detaching	$g_a - 1$	$g_m - 1$
crossover	g_m	g_b

Figure 5.8 & Table 5.4: A range of movement primitives in **case 2** of span-wm segment.

case 3 There are two movement primitives in this segment, *detaching* → *attaching*. Assuming g_m is a mid frame between g_a and g_b, a range of each movement primitive are given in Table 5.5.



Primitive	First	Last
detaching	$g_a - 1$	$g_m - 1$
attaching	g_m	$g_b + 1$

Figure 5.9 & Table 5.5: A range of movement primitives in **case 3** of span-wm segment.

case 4 Considering the segment shown in Figure 5.6, there only one movement primitive in the segment, *detaching*. Therefore, the first and last frame of this primitive are g_a and g_b respectively.

Once a range of each movement primitive are specified, the less of their skill parameters can be obtained by considering its first and last grasping postures.

5.2.2.2 Extracting Parameters of a Writhe Matrix

For each movement primitive, parameters of writhe matrices of its first and last grasping postures (frame) are used as its skill parameters. Parameters for each writhe matrix are different depending on its types, as described in Section 5.1.3.1. This section explains how to extract parameters from a given writhe matrix for both peak-wm and span-wm.

Extracting Parameters of Peak-wm For peak-wm T^p of size $n_1 \times n_2$, its parameters are referred to as $\mathcal{P}(T^p) = (w, p_x, p_y)$. Writhe is a sum of all elements in the matrix,

$$w = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} {T^p}_{i,j}. \tag{5.1}$$

The less of the parameters can be extracted by fitting the writhe matrix to the Bivariate Gaussian. The same method explained in Section 4.3.2.1 is used. The value of peak location are taken directly from the expectation value of the fitting result.

$$\boldsymbol{p} = (p_x, p_y) = (\mu_x, \mu_y). \tag{5.2}$$

This is also the incentive behind the name *peak* writhe matrix.

It can be seen from Equation (4.5) that the result of the Gaussian function fitting can be a partially fit in some circumstance. This means that the expectation value could be outside of the domain of writhe matrix, $(\mu_x, \mu_y) \notin [1, n_1] \times [1, n_2]$. In such cases, the values are projected back into the range before assigning to the peak location. This is done by:

- A line equation l represents a line that passes through (μ_x, μ_y) and having a slope θ (another parameter from the Gaussian fitting).
- A box b represents a rectangle that all of its side are the border of the range $[1, n_1] \times [1, n_2]$.
- Line l passes through b at points $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$. Peak location is assigned to be the point that is closer to the original (μ_x, μ_y).

Figure 5.10 visualises a method to project (μ_x, μ_y) before assigning to a peak.

In Table 5.1, it can be seen that the range of (μ_x, μ_y) is different from the range of (p_x, p_y). Therefore, the final value of $\boldsymbol{p}$ is changed from $[1, n_1] \times [1, n_2]$ to $[-1, 1] \times [-1, 1]$ by linearly scaling as

$$(p_x, p_y) \leftarrow \left(2 \cdot \frac{p_x}{n_1} - 1, 2 \cdot \frac{p_y}{n_2} - 1\right). \tag{5.3}$$

Extracting Parameters of Span-wm For span-wm T^s of size $n_1 \times n_2$, its parameters are referred to as $\mathcal{S}(T^s) = (w, \alpha, l_y)$. Writhe w is calculated using Equation (5.1). The less of the parameters is derived from the result of the Gaussian function fitting explained in Section 4.3.2.1. They are derived as the followings:

- A line equation l represents a line that passes through μ_x, μ_y and having a slope θ (another parameter from the Gaussian fitting).

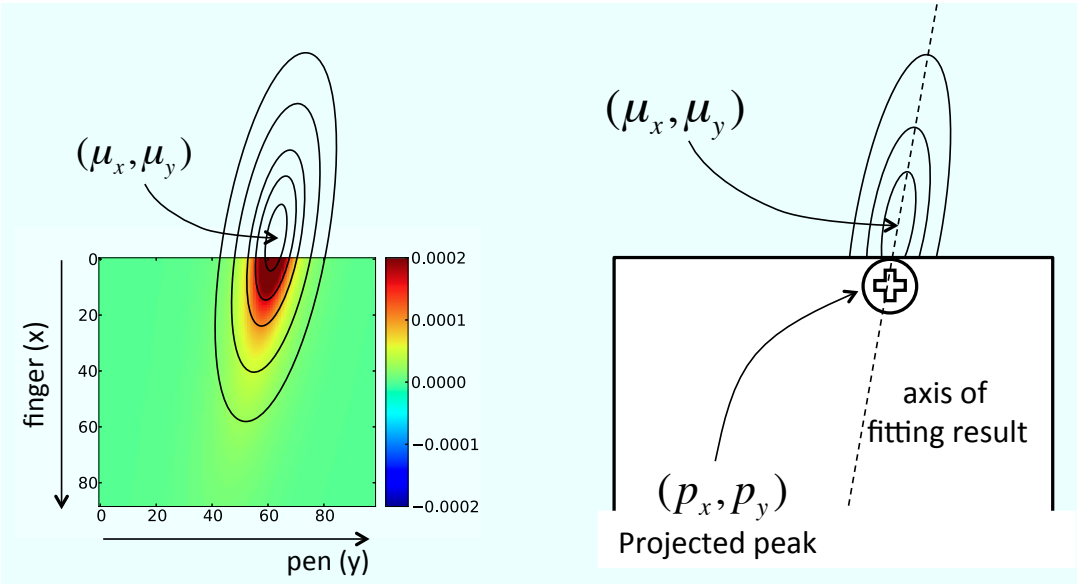


Figure 5.10: A Gaussian fitting result may result in the expectation value (μ_x, μ_y) that is outside of the domain of writhe matrix $[1, n_1] \times [1, n_2]$. The values are projected back into the range along an axis of the fitting before assigning to a peak.

- A span-line of span-wm is actually the same line as l. They have the same slope,

$$\alpha = \theta, \tag{5.4}$$

but with a different passing point. The other part l_y is defined as a y-coordinate of the point $(n_1/2, l_y)$, whom l passes through. Therefore,

$$l_y = \left(\frac{n_1}{2} - \mu_x\right) \tan\alpha + \mu_y. \tag{5.5}$$

This indicates that the location where the mid of the finger tangled with the object is considered as the average location on the object that is tangled by the whole finger.

Similarly with peak-wm, the range of the coordinate system change from $[1, n_1] \times [1, n_2]$ to $[-1, 1] \times [-1, 1]$. This effects in the linear scaling in l_y, and also the value of α as,

$$\alpha = \arctan\left(\frac{n_2}{n_1} \tan\theta\right). \tag{5.6}$$

5.2.2.3 Extracting Orientation of the Object

The orientation of the object is captured during human demonstration. This can be used directly as a skill parameter. An orientation of the object is supposed to be constant throughout a movement primitive. Therefore, the orientation of the pen of the first grasping posture (frame) is used. Alternatively, the average orientation of the first and last grasping posture, or of the whole sequence of grasping postures in the range of primitive can be used to handle the noise that might occur during demonstration.

5.3 Experimental Result

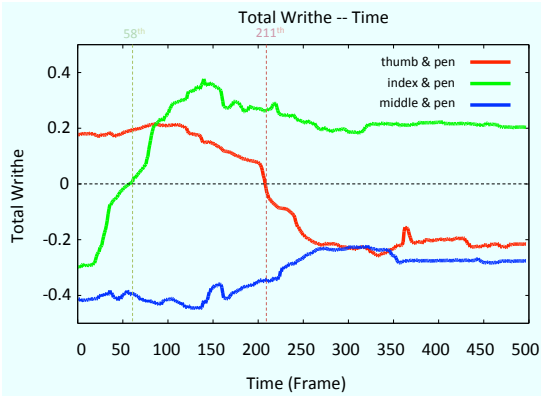
The same sequences of human regrasping movement considered in Section 4.4 are recognised against the task model. The segmentation method is used to divide the movement to many smaller segments depending on their types of writhe matrices. The recognition method uses this knowledge to determine movement primitives that occurs in the movement. Skill parameters for each movement primitives are extracted. The sequence of movement primitives, together with their skill parameters, are used to represent the observed regrasping movement.

Figure 5.11 shows a recognition result of Interdigital Step – Sequence 1. Total writhe value between each finger and the pen indicates the sign of all corresponding writhe matrices. At the particular moment, when a sign and a type of all three writhe matrices are known, a state of the grasping posture can be identified (e.g. $t_p^+ i_s^- m_p^-$ from frame 22^{nd} to 58^{th}). Using this information, movement primitives can be easily recognised. In the movement, there are six movement primitives. Three of which have an index finger (shown in green), while the others have a thumb (shown in red), as their major finger. Range of each movement primitive is calculated using method explain in Section 5.2.2.1. They indicate frames which the skill parameters of the movement primitives should be extracted. It is noticeable that there is no movement primitive connecting frame 87^{th} and 151^{st}. All grasping postures in the segment are considered to have the same state, i.e. $t_p^+ i_p^+ m_p^-$. Figure 5.12 illustrates corresponding grasping postures of initial and final frames of all movement primitives.

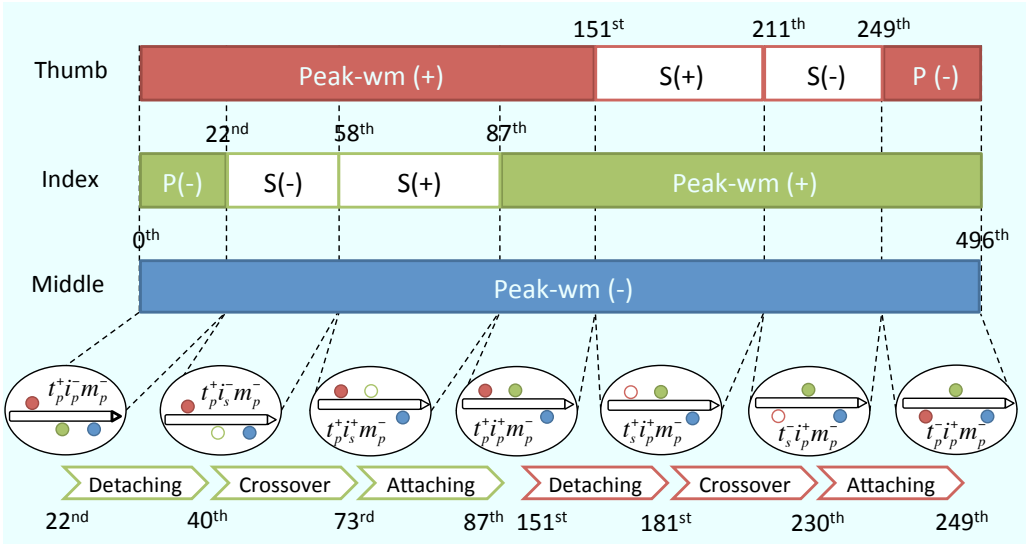
Similar recognition result of Interdigital Step – Sequence 2 is illustrated in Figure 5.13.

5.4 Summary of Chapter

A task model to recognise human grasping movement of a pen-like object is described. It is based on the classification of writhe matrices into two types: peak-wm and span-wm. A taxonomy of all grasping postures is described based on types and signs of writhe matrices between fingers and an object. In the



(a) Graph of total writhe. Three coloured line represent a writhe between pairs of strands: red= thumb and pen, green= index and pen, and blue= middle and pen. The red line crosses a zero value at frame 211^{th} and the green line crosses a zero value at frame 58^{th}.



(b) The regrasping movement is recognised into six movement primitives. The colour of the arrows indicate the major finger of the movement primitives (red for thumb and green for index finger). Frame numbers of the range of movement primitives are also illustrated.

Figure 5.11: Recognition results of Interdigital Step – Sequence 1.

taxonomy, there are limited numbers of transition between two grasping postures. Transitions are categorised into groups, which defines three movement primitives of the task model: *detaching*, *attaching*, and *crossover*. Skill parameters for each movement primitive are defined by states and parameters of writhe matrices of initial and final grasping postures. They are a necessary information needed to map each movement primitive onto a robotic hand.

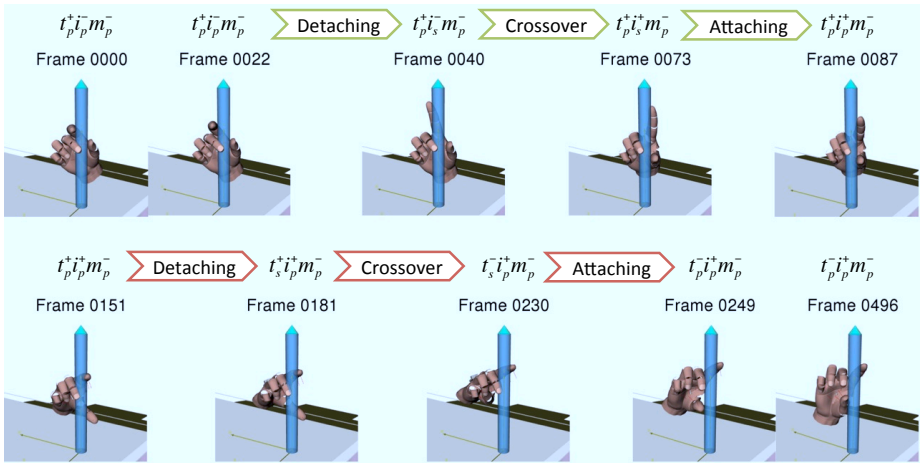
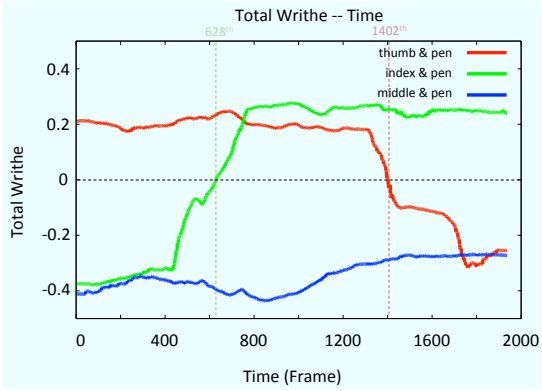
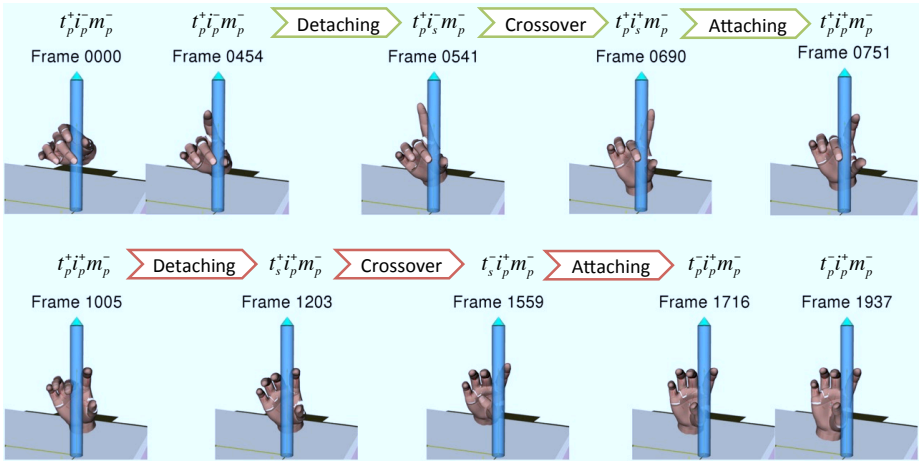


Figure 5.12: Grasping postures of the beginning and the end of all movement primitives are shown, together with the information on their states (Interdigital Step – Sequence 1).

A recognition method for the task model is also described in the chapter. Due to a strict definition of the task model, once human regrasping movement is segmented into changes of types of writhe matrices, movement primitives in the regrasping movement can be easily recognised. Their skill parameters are extracted from parameters describing writhe matrices, depending on their types. Finally, human regrasping movement is recognised and represented as a sequence of movement primitives together with their skill parameters.



(a) Graph of total writhe. Three coloured line represent a writhe between pairs of strands: red= thumb and pen, green= index and pen, and blue= middle and pen. The red line crosses a zero value at frame 1402^{nd} and the green line crosses a zero value at frame 628^{th}.



(b) Grasping postures of the beginning and the end of all movement primitives are shown, together with the information on their states.

Figure 5.13: Recognition results of Interdigital Step – Sequence 2.

CHAPTER

SIX

Mapping of Regrasping Movements

Contents

A method to map a human regrasping movement onto a robotic hand is described in this chapter. Knowledge obtained from a recognition of regrasping movement is used to guide a robot to imitate the movement. Unfortunately, a recognised skill parameters cannot be used directly to map the movement to a robotic hand. Therefore, a method to refine these skill parameters is necessary. Once refined, skill parameters are used directly in the mapping process. A regrasping movement is mapped onto a robotic hand on a movement primitive basis. In a RPO system, a movement of a robotic hand is generated for each movement primitive off-line. Then all the generated movements are connected

together to make a movement for the robotic hand that imitates the observed movement.

The chapter starts with an explanation of how to refine skill parameters obtained from the recognition. Then, a framework on how to map each type of movement primitives on to the robotic hand is described, followed by a method to connect them together. Finally, the chapter concludes experimental results of mapping a human regrasping movement onto a robotic hand in the real world environment.

6.1 Refinement of Skill Parameters

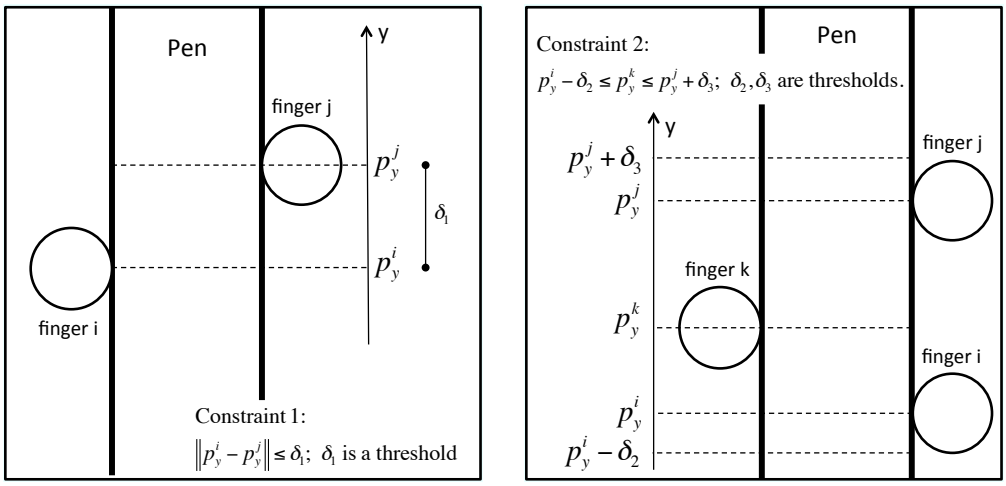
This section explains a method to refine skill parameters obtained from human demonstration. The main objective of this refinement is to modify skill parameters, so that they are suitable to be used to map the movement to a robotic hand. The refinement is necessary due to two main reasons. First, the inaccuracy of a data acquisition system dues to the demonstration in virtual environment (more details in Appendix A). This makes captured skill parameters violate some physical constraints. Second, the differences between structures of the human hand and the target robotic hand. This make the skill parameters without any refinement impossible to be used in the target robotic hand.

The section starts by introducing constraints that restrict skill parameters. Then a refinement method for each movement primitives based on these constraints are described.

6.1.1 Constraints for Skill Refinement

There are three constraints for skill refinement described in this subsection. The first two constraints are defined to force the skill parameters to obey the physical constraints when executing a movement with a robotic hand in the real world. The last constraint is defined to adjust skill parameters to be used in the target robotic hand. These constraints are defined based on the following preconditions:

- At least two fingers out of the three are in contact with the object, while contacts are modeled as a soft finger frictional contact [Kao *et al.* 2008].
- an object is a pen-like object.
- Within each movement primitive, there is no movement of the object.
- The problem is simplified into two dimensional problem.
- There are only some specific area on the fingers of robotic hand that can be contacted with the object.



(a) Visualisation of **Constraint 1** (b) Visualisation of **Constraint 2**

Figure 6.1: Visualisation of constraints that are related to contact locations.

These preconditions lead to the following constraints.

Constraint 1 When only two fingers are in contact with the pen, their contact location on the pen must be on the opposite side of the pen and near to each other, in order to kept the pen in stable state and not moving. Considering $\mathcal{P}_i(w^i, p_x^i, p_y^i)$ and $\mathcal{P}_j(w^j, p_x^j, p_y^j)$ as the skill parameters of two fingers that are in contact with the pen, these constraints can be written as

$$\|p_y^i - p_y^j\| \le \delta_1, \tag{6.1}$$

where $\delta_1 \ge 0$ is a small threshold value.Figure 6.1(a) illustrates the constraint in a better visualisable manner.

Constraint 2 When all three fingers are in contact with the pen, two of the contact will be in one side of the pen while another contact will on the other side. A contact location on the pen of the lone contact must be in between the others. Considering $\mathcal{P}_i(w^i, p_x^i, p_y^i)$, $\mathcal{P}_j(w^j, p_x^j, p_y^j)$ and $\mathcal{P}_k(w^k, p_x^k, p_y^k)$ as the skill parameters of all fingers that are in contact with the pen where $\mathcal{P}_k$ is a lone contact and $p_y^i < p_y^j$ is assumed without loss of generality, this constraint can be written as

$$p_y^i - \delta_2 \le p_y^k \le p_y^j + \delta_3, \tag{6.2}$$

where $\delta_2, \delta_3 \ge 0$ are small threshold values. Figure 6.1(b) shows more comprehensive illustration of this constraint.

Constraint 3 For a particular robotic hand, the area on fingers that can be in

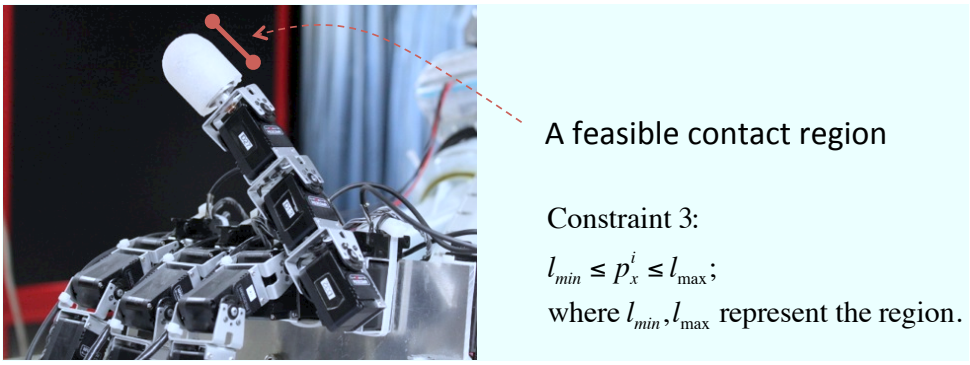


Figure 6.2: Limited area of contact on a robotic hand.

contact with the pen can be limited. The related skill parameters must be adjusted accordingly. Considering $\mathcal{P}_i(w^i, p_x^i, p_y^i)$ as skill parameters of the finger that is in contact, and $[l_{min}, l_{max}] \in [-1, 1]$ are the area of the robotic finger that could be in contact, this constraint can be written as

$$l_{min} \leq p_x^i \leq l_{max}. \tag{6.3}$$

6.1.2 Skill Refinement for Movement Primitives

A method for refining skill parameters of each movement primitive is based on preconditions and constraints explained in Section 6.1.1. Constraints are applied to the initial and final writhe matrices of the movement primitives to initially modify their skill parameters. Then a different refinement method is applied depending on a type of the movement primitive.

For every type of movement primitive, both initial or final grasping postures has three writhe matrices involved as described in Table 5.2: one for major finger and two for two minor fingers. Without loss of generality, assuming that

- $\mathcal{F}_0(w^0, \ldots)$ is parameters of writhe matrix (either peak-wm or span-wm) of the major finger m_0,
- $\mathcal{P}_1(w^1, p_x^1, p_y^1)$ is parameters of peak-wm of the first minor finger m_1 where $w^0 w^1 > 0$ (same sign of writhe value),
- and $\mathcal{P}_2(w^2, p_x^2, p_y^2)$ is parameters of peak-wm of the second minor finger m_2 where $w^0 w^2 < 0$ (opposite sign of writhe value),

constraints modify these parameters as the following:

- if $\mathcal{F}_0$ is parameters of span-wm, **Constraint 1** is applied by modifying

$$p_y^2 = \begin{cases} p_y^2 & \|p_y^2 - p_y^1\| \leq \delta_1 \\ p_y^1 - \delta_1 & p_y^2 < p_y^1 - \delta_1 \\ p_y^1 + \delta_1 & p_y^2 > p_y^1 + \delta_1 \end{cases}, \tag{6.4}$$

  where $\delta_1 \leq 0$ is a small threshold value.

- if $\mathcal{F}_0$ is parameters of peak-wm and $p_y^0 < p_y^1$ is assumed without loss of generality, **Constraint 2** is applied by modifying

$$p_y^2 = \begin{cases} p_y^2 & p_y^0 - \delta_2 \leq p_y^2 \leq p_y^1 + \delta_3 \\ p_y^0 - \delta_2 & p_y^2 < p_y^0 - \delta_2 \\ p_y^1 + \delta_3 & p_y^2 > p_y^1 + \delta_3 \end{cases}, \tag{6.5}$$

  where $\delta_2, \delta_3 \geq 0$ are small threshold values.

- Notice that **Constraint 3** is applied to any peak-wm of any finger to modify a contact location on the finger to be feasible for a target robotic hand.

For each type of movement primitive, skill parameters are further refined so that when the movement is mapped to the robotic hand, it does not affect on the movement of the pen. Fundamentally, the refinement methods are based on the assumption that if the contact location on the pen of both minor fingers are the same in both initial and final grasping postures and the location of the pen is fixed when the movement primitive is interpolated for the robotic hand, a location (position and orientation) of the pen should not change much during the real execution. Therefore, it depends on whether to use the contact location of the initial or the final grasping posture for both minor fingers. These refinement methods are different for each type of movement primitive, which are described below.

- For *Detaching*, contact locations on the pen of the two minor fingers of the final grasping posture are used. The contact locations on the pen the initial grasping posture are then changed and refined to these values.

- For *Crossover* and *Attaching*, contact locations on the pen of the two minor fingers of the initial grasping posture are used. The contact locations on the pen the final grasping posture are then changed and refined to these values.

Note that the refinement is applied sequentially by the order of movement primitives. For two consecutive movement primitives that share common grasping posture and skill parameters, the changes in skill parameters of the prior movement primitive will affect the changes of the skill parameters of the latter movement primitive.

6.2 Mapping of Movement Primitives

The section describes a framework to map movement primitives to a robotic hand. The purpose of this mapping framework is to generate trajectories of joints/hand of the robotic hand that is resembled to a demonstrated movement. Skill parameters of each movement primitive are used as a fundamental knowledge to ensure that the generated movement primitive is similar to the demonstrated one.

Generating (or interpolating) a trajectory of the robotic hand can be considered as solving a series of Inverse Kinematics (IK) problem. In short, a time-series of the goal constraints is specified. Then a hand configuration and location are synthesised to meet the constraints of each frame by solving an IK problem. In a general IK problem, the goal constraints are usually a desired trajectory of the end effector, which could be yielded from any motion planning technique. Buss presented an extensive survey article on the techniques used to solve the IK problem [Buss 2004].

In this framework, the tangle relation of hand and object are used as a goal constraints. To be more specific, the trajectory of desired writhe matrices are used to lead the interpolation of the robotic hand. The method to generate the trajectory of desired writhe matrices that connects between two writhe matrices is also proposed in the framework. In fact, the idea of planning a trajectory in topological space was originally used to synthesise a whole body motion of a character [Ho & Komura 2009a]. The method is adopted and modified to be used in this framework.

The following priors and inputs are given to the system.

- A structure of the robotic hand, and an exact geometry of the pen.
- Skill parameters of the movement primitive.
- Location (position and orientation) of the pen; more details in Section 6.3.
- Initial location (position and orientation) and configuration of robotic hand for both initial and final state described in the skill parameters; more details in Section 6.2.2.

To generate a trajectory of a robotic hand for each movement primitive, the following steps are used.

step 1 Mapping skill parameters to the initial and final robot grasping postures. This step maps an abstract information obtained from human demonstration to the robotic hand. It creates the initial and final grasping postures of the robotic hand that have the same skill parameters as the movement primitive.

step 2 Interpolating the robotic hand from its initial grasping posture to its final grasping posture. This step generates a trajectory of the robotic hand to connect between its initial and final grasping postures. The motion is synthesised by planning a trajectory in the topology space.

The less of the section is organized as the followings. First, the method for solving IK problem of the robotic hand in the topology space is described. It is used heavily in both steps for the mapping framework. Then both steps of the mapping framework are explained respectively.

6.2.1 Inverse Kinematics in Topology Space

The method is first used by Ho and Komura [Ho & Komura 2009a] to solve IK problem for a character. This section describes the method and how to formulate it to solve an IK problem for a robotic hand.

A generalized coordinate of the robotic hand, $\mathbf{r}$ at the specific moment is described by its joint angles and its location (position and orientation). Writhe matrix T represents tangle relation between the robotic hand and the object. It represents either a writhe matrix between a specific finger and the object, or a concatenation of them. To simplify the representation, let $\mathbf{t}$ be a vector representation of writhe matrix T. For writhe matrix $T_{m\times n}$, $\mathbf{t}$ is a vector of size $m \times n$ where,

$$\mathbf{t} = [T_{(1,1)}, \ldots, T_{(1,n)}, T_{(2,1)}, \ldots, T_{(2,n)}, \ldots, T_{(m_1)}, \ldots, T_{(m,n)}]^\intercal \tag{6.6}$$

Assume a location of the object is fixed, writhe matrix is a function of a generalized coordinate of the robotic hand alone; this fact can be expressed as $\mathbf{t} = \mathbf{t}(\mathbf{r})$, where $\mathbf{t}$ is a

Given a target tangle relation between the robotic hand and the object as writhe matrix $\mathbf{t}^d$, the IK problem is to find a hand generalized coordinate, $\mathbf{r}$ so that

$$\mathbf{t}^d = \mathbf{t}(\mathbf{r}). \tag{6.7}$$

An iterative method is used to approximate a solution for Equation (6.7). A Jacobian matrix is used to linearly approximate the function of $\mathbf{t}$. The Jacobian matrix J of function $\mathbf{t}$ is a function of vector $\mathbf{r}$ and defines as,

$$J(\mathbf{r}) = \frac{\partial \mathbf{t}}{\partial \mathbf{r}} = \begin{bmatrix} \frac{\partial t_1}{\partial r_1} & \cdots & \frac{\partial t_1}{\partial r_{|\mathbf{r}|}} \\ \vdots & \ddots & \vdots \\ \frac{\partial t_{|\mathbf{t}|}}{\partial r_1} & \cdots & \frac{\partial t_{|\mathbf{t}|}}{\partial r_{|\mathbf{r}|}} \end{bmatrix}. \tag{6.8}$$

The small change in the generalized coordinate of the robotic hand relates to the

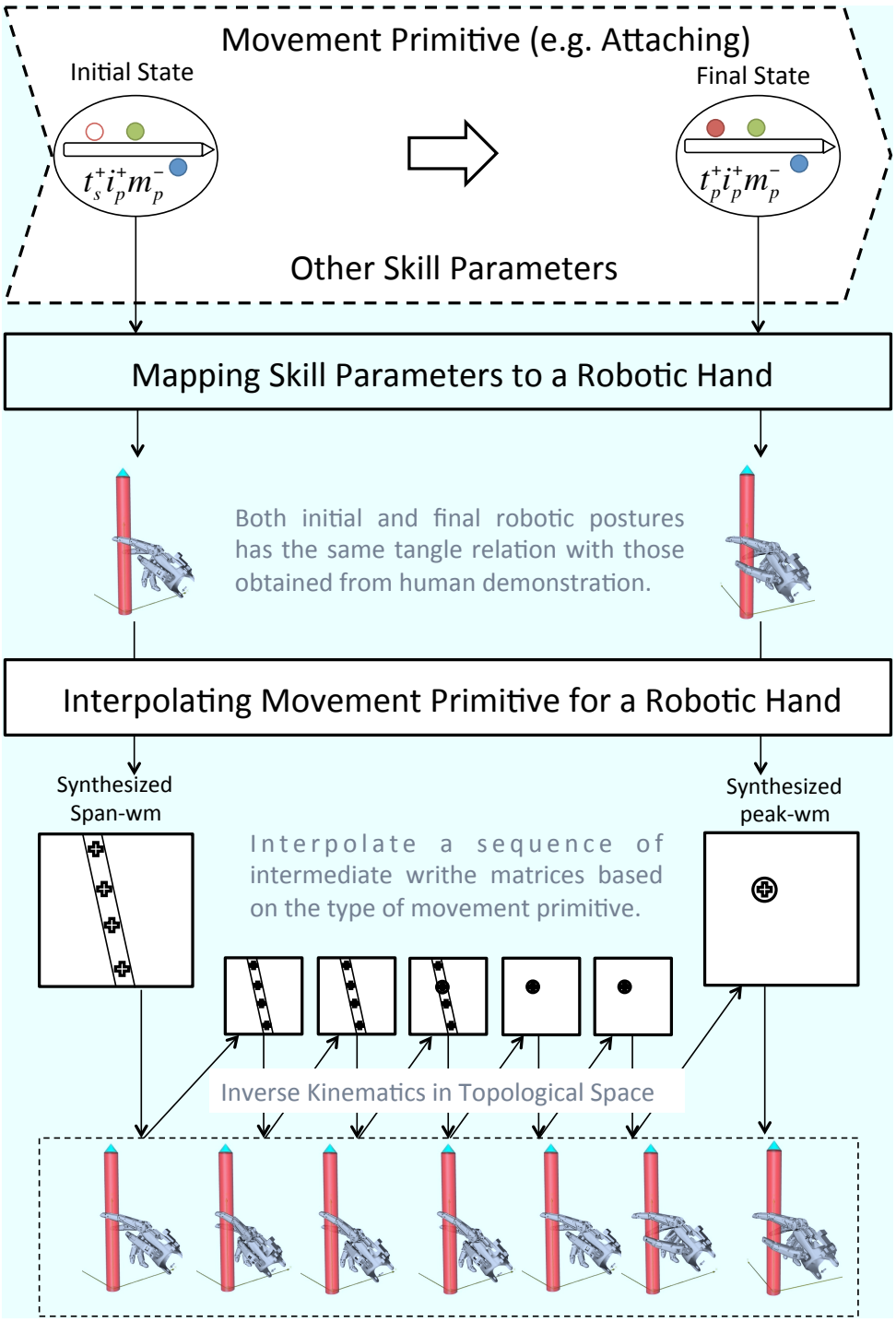


Figure 6.3: Framework for mapping movement primitives to a robotic hand. A movement primitive is passed to a system together with its skill parameters obtained from human demonstration. A corresponding skill parameters is first mapped to a robotic hand to create initial and final grasping postures. Then a trajectory of a robotic hand is created to connect between the two grasping postures by interpolating in a topological space.

change in writhe matrix through this Jacobian matrix as,

$$\Delta \mathbf{t} = J(\mathbf{r})\, \Delta \mathbf{r}. \tag{6.9}$$

Starting with a generalized coordinate of the robotic hand $\mathbf{r}$ with a writhe matrix $\mathbf{t}$ and a target writhe matrix $\mathbf{t}^d$, the key idea of solving this IK problem is to update a generalized coordinate with $\Delta \mathbf{r}$ so that it effects in a change of writhe matrix $\Delta \mathbf{t}$ which is approximately equal to $\mathbf{t}^d - \mathbf{t}$.

To find the most appropriate value of $\Delta \mathbf{t}$, the damped least squares method [Wampler 1986, Nakamura & Hanafusa 1986] is used. The objective function to search for $\Delta \mathbf{t}$ is given as the following:

$$\underset{\Delta \mathbf{r}}{\text{minimize}} \qquad \|J(\mathbf{r})\, \Delta \mathbf{r} - (\mathbf{t}^d - \mathbf{t})\|^2 + \lambda^2 \|\Delta \mathbf{r}\|^2, \tag{6.10}$$

where $\lambda \in \mathbb{R}$ is a non-zero damping constant. Damping constant λ can also be chosen to be different for each finger or each element in the generalized coordinate.

Other constraints can be included in Equation (6.10) in order to restrain the generated regrasping movement. In this system, two additional constraints are added; a collision avoidance and a kinematic limitation of the robot hand.

Collision between a robotic hand and manipulated object can be avoided by limiting the maximum writhe between segments of concerned strands. When two line segments approach each other, the absolute value of their writhe increases and becomes 0.5 at the moment of crossing. This can be described as :

$$\|J(\mathbf{r})\, \Delta \mathbf{r} + \mathbf{t}\| \leq \sigma, \tag{6.11}$$

where σ is a threshold vector.

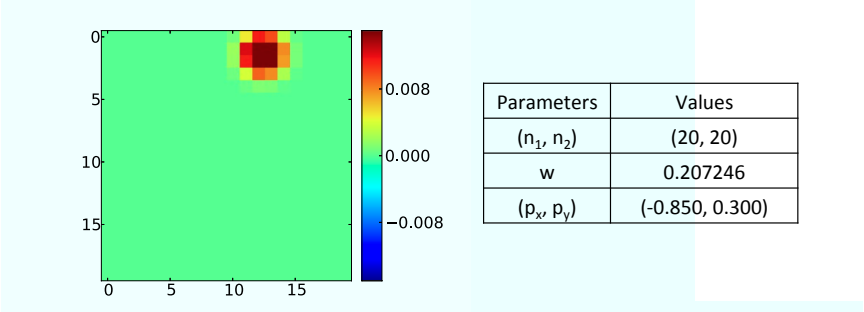
Kinematic limitation of finger/hand are restricted by assigning a feasible search space for $\Delta \mathbf{r}$. Assume that $\mathbf{r}_{min,k}$ and $\mathbf{r}_{max,k}$ are a minimum and maximum possible value for joints and movement of robot hand, the constraint can be written as :

$$\mathbf{r}_{min} - \mathbf{r} \leq \Delta \mathbf{r} \leq \mathbf{r}_{max} - \mathbf{r}. \tag{6.12}$$

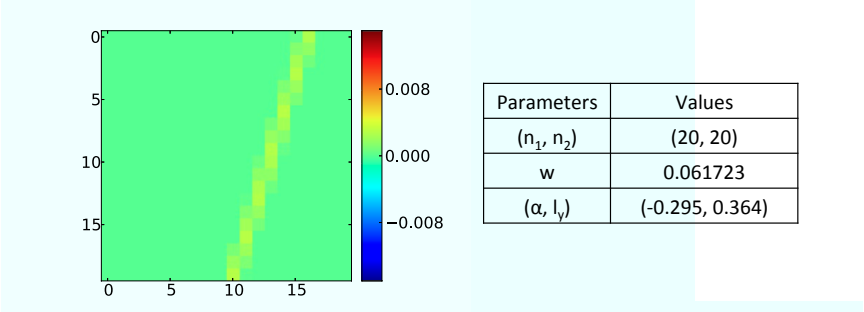
To sum up, to approximate a solution for Equation (6.7), the following objective function and constraints are minimized,

$$\begin{aligned} \underset{\Delta \mathbf{r}}{\text{minimize}} \quad & \|J(\mathbf{r})\, \Delta \mathbf{r} - (\mathbf{t}^d - \mathbf{t})\|^2 + \lambda^2 \|\Delta \mathbf{r}\|^2 \\ \text{subject to} \quad & \|J(\mathbf{r})\, \Delta \mathbf{r} + \mathbf{t}\| \leq \sigma \\ & \mathbf{r}_{min} - \mathbf{r} \leq \Delta \mathbf{r} \leq \mathbf{r}_{max} - \mathbf{r}. \end{aligned} \tag{6.13}$$

The problem is formulated as a quadratic function of $\Delta \mathbf{r}$, subject to its linear constraints, widely known as a Quadratic Programming (QP) problem [Nocedal



(a) Synthesised peak-wm



(b) Synthesised span-wm

Figure 6.4: Examples of synthesised writhe matrices.

& Wright 2006]. The generalized coordinate $\Delta\mathbf{r}$ are iteratively updated so that the tangle relation, a writhe matrix in this case, is sufficiently close to a target writhe matrix $\mathbf{t}_d$.

Note that the Jacobian matrix defined in Equation (6.8) is numerically estimated because Equation (6.7) is very complex and cannot easily be written in a closed form. Each element of the Jacobian is calculated as

$$\frac{\partial t_i}{\partial r_j} = \frac{t_i\left(r_1, \ldots, r_j + h, \ldots, r_{|\mathbf{r}|}\right) - t_i\left(\mathbf{r}\right)}{h}, \tag{6.14}$$

where $1 \leq i \leq |\mathbf{t}|$, $1 \leq j \leq |\mathbf{r}|$ and the value of h close to zero.

6.2.1.1 Synthesising Writhe Matrix from its Parameters

Skill parameters obtained from human demonstration is passed to the robot system in the form of parameters of writhe matrices, $\mathcal{P}$ or $\mathcal{S}$. In order to use these informations for robot mapping, it is necessary that these parameters are converted back to the writhe matrix so that the method to solve IK problem in Section 6.2.1 can be used. This subsection describes a method to synthesise each type of writhe matrices from their corresponding parameters.

Synthesising Peak-wm Given parameters of peak-wm $\mathcal{P}$ described in Table 5.1, a synthesised writhe matrix ${}^{\mathcal{P}}T$ of size $n_1 \times n_2$ can be synthesised using the following procedures:

1. Scaling (p_x, p_y) linearly from the range of $[-1, 1] \times [-1, 1]$ to $[1, n_1] \times [1, n_2]$, and referred to them as (p'_x, p'_y)

2. Using a (singular) Bivariate Gaussian function to create each element of the target writhe matrix. This can be mathematically written as,

$$ {}^{\mathcal{P}}T'_{i,j} = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\left(\frac{(i-p'_x)^2}{2\sigma_x^2} + \frac{(j-p'_y)^2}{2\sigma_y^2}\right)\right), \tag{6.15} $$

   $\forall (i, j) \in [1, n_1] \times [1, n_2]$. Value of σ_x and σ_y effect the distance between strands of the robotic hand and the object when they are in contact. These values can be set empirically for each pair of stands representing a finger and a object.

3. Scaling a writhe matrix ${}^{\mathcal{P}}T'$ to a total writhe value w in $\mathcal{P}$ including the sign.

$$ {}^{\mathcal{P}}T_{i,j} = w \frac{{}^{\mathcal{P}}T'_{i,j}}{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} {}^{\mathcal{P}}T'_{i,j}}. \tag{6.16} $$

Notice that a singular Bivariate Gaussian function is used in Equation (6.15). This is because when the finger is in contact with the object, the rotation angle θ of the original writhe matrix is not significant. Figure 6.4(a) illustrates an example of synthesised peak-wm.

Synthesising Span-wm Given parameters of span-wm $\mathcal{S}$ described in Table 5.1, a synthesised writhe matrix ${}^{\mathcal{S}}T$ of size $n_1 \times n_2$ can be synthesised using the following procedures:

1. Scaling (α, l_y) from the range of $[-1, 1] \times [-1, 1]$ to $[1, n_1] \times [1, n_2]$ by inversing the method explained in Section 5.2.2.2, and referred to them as (α', l'_y).

2. Considering a line equation $l : j = i \tan(\alpha') + l'_y$, each element of the writhe matrix is assigned a constant value c according to whether the line l passes through it. In most cases, the line l will not pass through an integer point $(i, j) \in \mathbb{Z} \times \mathbb{Z}$. Therefore, one of i or j will be assigned to be an integer and be iterated over it. Assuming $i \in \mathbb{Z}$, a point (i, j_r) is passed through

Algorithm 2: SynSpanWM Synthesised Span Writhe Matrix

Input: (α', l'_y) Scaled span-line of a desired span-wm.
Input: (n_1, n_2); Size of a target writhe matrix.
Output: ${}^{\mathcal{S}}T'$; Writhe matrix of size $n_1 \times n_2$, synthesised from the span-line (α', l'_y)

```
% Initialising ^S T' with zero-matrix of size n_1 × n_2
^S T' ← zero[n_1][n_2]
% Deciding whether to iterate over i or j
if tan(α') ≤ n_2/n_1 then
    % Iteration over i ∈ [1, n_1] ∈ ℤ
    for i ← 1..n_1 do
        j_r ← i tan(α') + l'_y
        % Check whether j_r is an integer.
        if j_r ∈ ℤ then
            % Assigning an element with a constant c = 1.
            ^S T'_{i,j_r} ← 1
        else
            % Assigning elements on both nearest blocks,
            % proportionated to their distance to (i, j_r).
            ^S T'_{i,floor(j_r)} ← (j_r − floor(j_r))
            ^S T'_{i,ceil(j_r)} ← (ceil(j_r) − j_r 1)
else
    % Iteration over j ∈ [1, n_2] ∈ ℤ
    for j ← 1..n_2 do
        % Similar method to the iteration over i
        ...
return ^S T'
```

by line l. If $j_r \notin \mathbb{Z}$, both $(i, floor(j_r))$ and $(i, ceil(j_r))$ are assigned with a fraction of the constant value c proportionated to their distance to (i, j_r).

The decision whether to iterate over an integer i or j is depended on the value of α'. Considering a line l_d that passes through point $(1, 1)$ and (n_1, n_2) which represent a limit of the domain of writhe matrix, the iteration will be over an integer i when a slope of line l is less slanted than a slope of line l_d : $\tan(\alpha') \leq {}^{n_2}/_{n_1}$ and over an integer j otherwise.

The writhe matrix that is synthesised by above method is referred to as ${}^{\mathcal{S}}T'$. Algorithm 2 describes the method more precisely, while Figure 6.5 illustrates the method more visually.

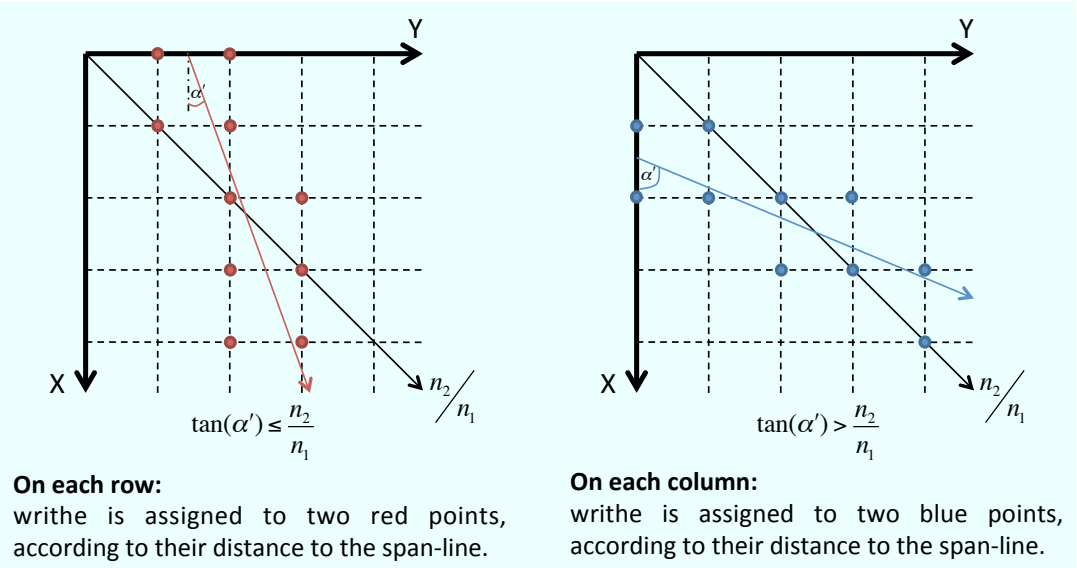


Figure 6.5: Two span-wms are synthesised differently based on the orientation (α') of their span-line.

3. Scaling a writhe matrix ${}^{\mathcal{S}}T'$ to a total writhe value w in $\mathcal{S}$ including the sign using a Equation (6.16).

Figure 6.4(b) illustrates an example of synthesised span-wm.

Size of Synthesised Writhe Matrix Size of Synthesises writhe matrix is directly related to the granularity of how the strands of the robotic hand and the object are divided into line segments The more the number of line segments is divided, the more the number of elements in writhe matrix will be used to solve the IK problem explained earlier in the section. This is similar to the discussion in Section 4.3.2.1. However, more line segments does not guarantee a better result in solving IK problem. It could also be a trade-off to a computation time because it increases the number of variables in the optimization problem. The number of line segments are identified empirically during the experiment; however, the number of line segments are considerably low (around 10 times less) comparing to the case used during recognition.

Notice that each element of writhe matrix is treated equally during synthesis. This means that all line segments that represent same strand should have the same (or at least similar) segment length. Algorithm 1 is also used for this purpose.

There is also a reason why some parameters of both types of writhe matrix are scaled to the range of $[-1, 1] \times [-1, 1]$. The size of writhe matrix during a recognition process and the size of writhe matrix during mapping process might

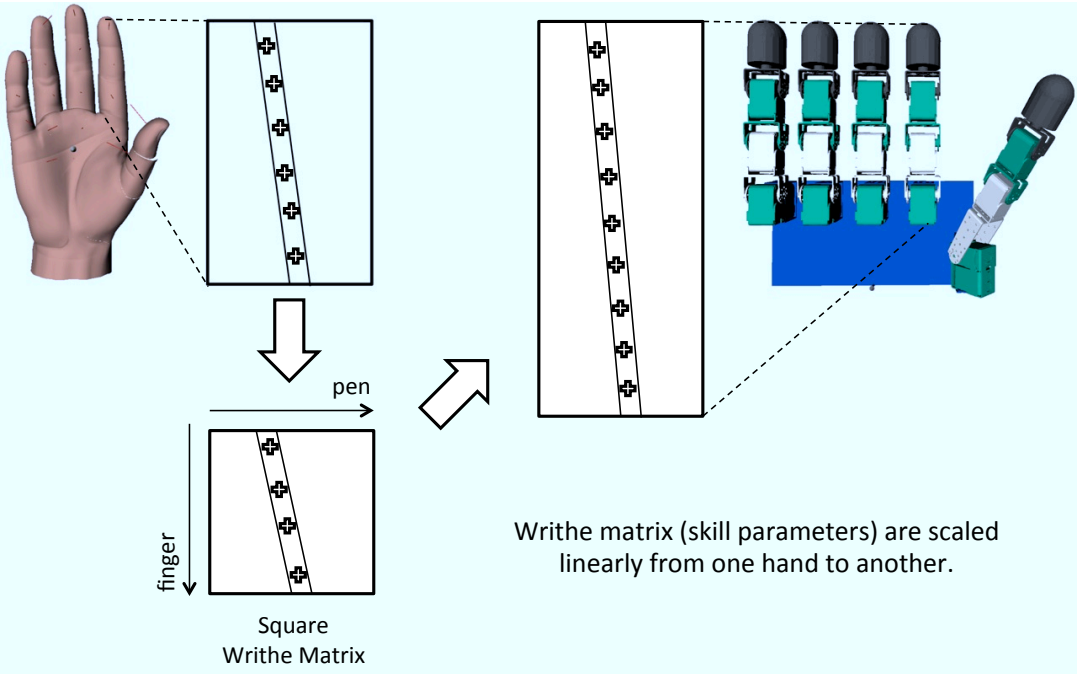


Figure 6.6: Mapping of writhe matrices for hands with different fingers' length. Skill parameters are scaled to standard length before scaling back to the length of the destination finger. They are treated equally regardless of their structure (e.g. lengths of finger segments.)

be different. Scaling to a standard length makes the mapping between the two possible. However, this mean that the finger of human and a target robotic hand are mapped together regardless of their length and structure, as shown in Figure 6.6.

6.2.2 Mapping Skill Parameters to Robot Postures

Before generating a trajectory of a robotic hand for each movement primitive, initial and final grasping postures of the robotic hand are created to have the same skill parameters as of the movement primitive. This can be done by solving the IK problem, individually for initial and final postures, given parameters of their writhe matrices which are part of the skill parameters explained in Section 6.2.1. The remaining question is how to initialise the robotic grasping posture so that it can be used as an initial generalized coordinated of the IK problem.

The initial grasping posture for IK problem can be divided into the initialization of two parts: pen location, and a configuration and location of the robotic hand. The former can be initialised using an arbitrary position with a parameter v_b from skill parameters directly. For the latter, a configuration of the robotic hand for IK problem is initialised with state information i.e. $s_i, or s_f$ in Table 5.2.

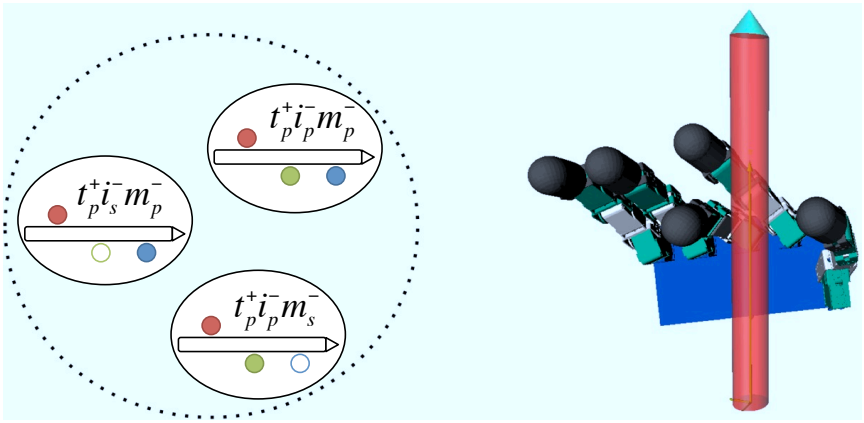


Figure 6.7: An initial robotic grasping posture for all states with $t^+i^-m^-$. It is used to initialise a robotic configuration before it is mapped to skill parameters obtained from human demonstration.

For each group of states with the same sign of writhe matrices for all three fingers, i.e. those in each dashed line circle, a robotic grasping posture is prepared and given as an input to the system. The grasping posture does not need to be in a particular type of writhe matrix or to have a specific skill parameters of the writhe matrices. It is only needed to have the same correspondence of the sign of all writhe matrices with states it represented. This will provide an initial configuration of the robotic hand and also a relative location of the hand to the pen, which can then be adjusted accordingly to the location of the pen. Figure 6.7 shows examples of an initial grasping posture for states with $t^+i^-m^-$.

Notice that in this step, the trajectory to reach the desired writhe matrices is not important. The hand configuration and location are updated iteratively using Equation (6.13) until writhe matrices between the robotic hand and then pen are similar to the desired writhe matrices.

6.2.3 Interpolate Hand Motion between Initial and Final States

Once initial and final grasping postures of a robotic hand for the movement primitive is created, the next step is to generate a trajectory of the robotic hand that connects them together. An IK solver explained in Section 6.2.1 is used to generate the trajectory of the robotic hand. The method tries to move the robotic hand so that its writhe matrices are similar to the desired writhe matrices. Therefore, to interpolate a trajectory for the robotic hand, a sequence of intermediate writhe matrices to lead the trajectory interpolation is required. In other words, this sequence of intermediate writhe matrices dictates how the movement of robot hand turns out to be for each movement primitives. In this

section, a method to create a sequence of intermediate writhe matrices for each movement primitives is explained.

A sequence of intermediate writhe matrices of major finger for each movement primitive is created as follows:

- *Detaching* : Consider $\mathcal{P}_i(w^i, p_x^i, p_y^i)$ and $\mathcal{S}_j(w^j, \alpha^j, l_y^j)$ as skill parameters for initial peak-wm and final span-wm writhe matrix. In this case, a mid span-wm writhe matrix is necessary. It is assigned with skill parameters $\mathcal{S}_m(w^m, \alpha^m, l_y^m)$, where

$$\begin{aligned} w^m &= w^j \\ \alpha^m &= \alpha^j \\ l_y^m &= p_y^i - \alpha^j p_x^i. \end{aligned} \tag{6.17}$$

  A pair (α^m, l_y^m) represents a span-line that pass through point (p_x^i, p_y^i), but has the same direction as a span-line (α^j, l_y^j). Then a process of creating a sequence of intermediate writhe matrices for this movement primitives is divided into two steps. First, intermediate writhe matrices between $\mathcal{P}_i$ and $\mathcal{S}_m$ are created. This can be done by synthesising both writhe matrices, T^i, T^m,from their skill parameters, and then element-wise linearly interpolating between them to create all intermediate writhe matrices. Assume that $T^r, r \in [1, n]$ is one of the n intermediate writhe matrices that would be created between T^i and T^m, T^r of size $n_1 \times n_2$ is calculated as

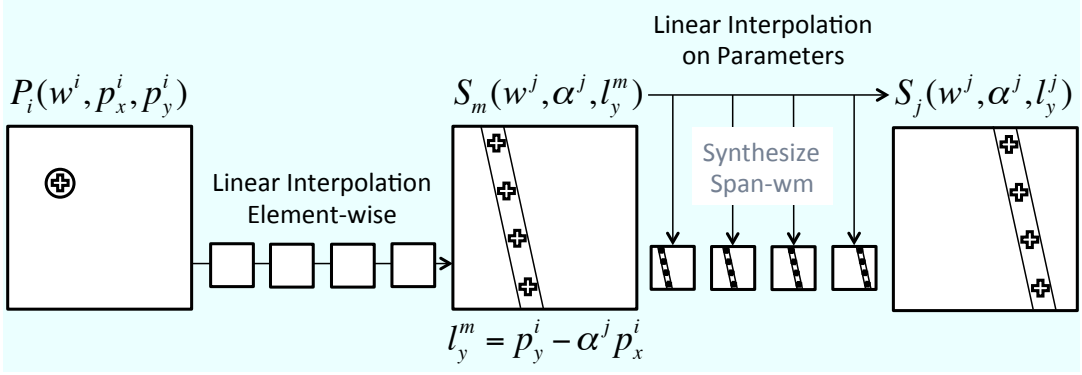
$$T_{p,q}^r = r\frac{T_{p,q}^m - T_{p,q}^i}{n+1} + T_{p,q}^i \tag{6.18}$$

  $\forall (p, q) \in [1, n_1] \times [1, n_2]$. Second, intermediate writhe matrices between $\mathcal{S}_m$ and $\mathcal{S}_j$ by linearly interpolating between their skill parameters to obtain skill parameters for all intermediate span-wm, $\mathcal{S}_s$. Then each intermediate writhe matrix is synthesised from this parameter $\mathcal{S}_s$.

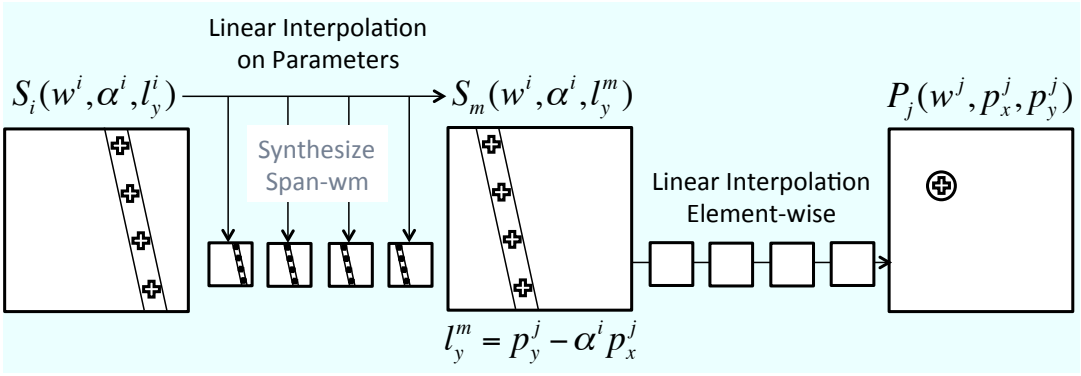
- *Attaching* : It uses the same algorithms as *Detaching*, but in the reversed direction.

- *Crossover* : Consider $\mathcal{S}_i(w^i, \alpha^i, l_y^i)$ and $\mathcal{S}_j(w^j, \alpha^j, l_y^j)$ as skill parameters for initial span-wm and final span-wm writhe matrix. Mid span-wm is also necessary in this case. It is assigned with skill parameters $\mathcal{S}_m(w^m, \alpha^m, l_y^m)$, where

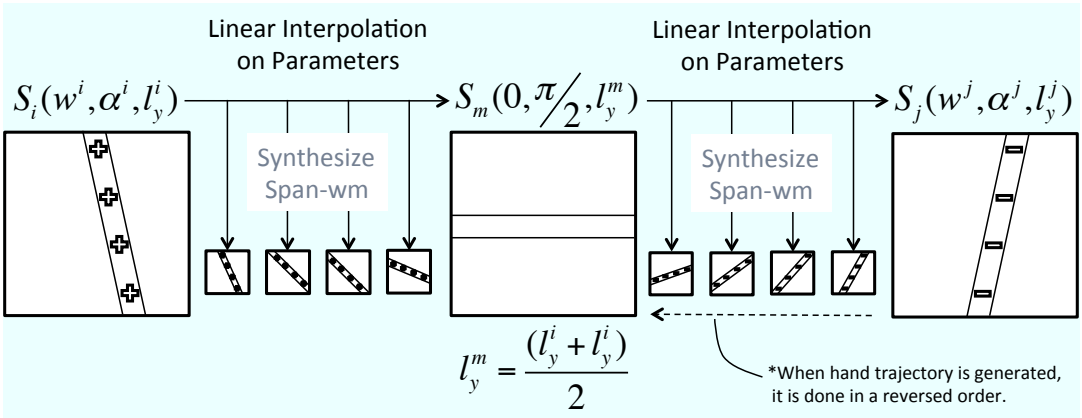
$$\begin{aligned} w^m &\approx 0 \\ \alpha^m &= \pi/2 \\ l_y^m &= (l_y^i + l_y^j)/2. \end{aligned} \tag{6.19}$$



(a) *Detaching*. The process is divided into two steps, connecting by a mid span-wm S_m. The mid span-wm has same writhe as the final span-wm ($w^m = w^j$), same orientation of the span-line as the final span-wm ($\alpha^m = \alpha^j$), but its span-line pass through a peak of the initial peak-wm ($l_y^m = p_y^i - \alpha^j p_x^j$). Firstly, intermediate writhe matrices between the initial peak-wm and the mid span-wm are created by linearly interpolating each element of their synthesised writhe matrices. Secondly, intermediate writhe matrices between the mid span-wm and the final span-wm are created by linearly interpolating between their parameters and then synthesised span-wm based on the interpolated parameters.



(b) *Attaching*. This is simply a reversal of the method of *detaching* movement primitive.



(c) *Crossover*. The process is divided into two steps, connecting by a mid zero-wm S_m. The mid zero-wm is a special kind of span-wm that its writhe is close to zero ($w^m \approx 0$) and its span-line is a horizontal line ($\alpha_m = {}^{pi}/_2$). Note that l_y^m is set to $(l_y^i + l_y^j)/2$ for connecting purposes. Both steps generate intermediate writhe matrices by linearly interpolating between their parameters and then synthesised span-wm based on the interpolated parameters.

Figure 6.8: Methods to generate a sequence of intermediate writhe matrices for a major finger for each movement primitives.

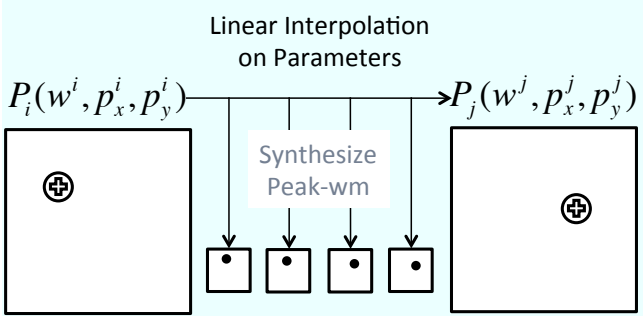


Figure 6.9: Methods to generate a sequence of intermediate writhe matrices for minor fingers for all movement primitives. Intermediate writhe matrices between an initial peak-wm and a final peak-wm are created by linearly interpolating between their parameters and then synthesised peak-wm based on the interpolated parameters.

This $\mathcal{S}_m$ represents a zero-wm, a special kind of span-wm, where α^m is a slop of span-line that results in the horizontal line in the writhe matrix. In the similar manner as *detaching*, the process of creating a sequence of intermediate writhe matrix is divided into two steps: creating intermediate writhe matrices between S_i and S_m and creating intermediate writhe matrices between S_m and S_j. Both steps can be done by linearly interpolating between skill parameters of initial and final writhe matrices to obtain skill parameters for all intermediate span-wm writhe matrices. Then, each intermediate writhe matrix is synthesised from its skill parameters.

Notice that the process of generate a trajectory of the robotic hand for *crossover* is also divided into two steps. The first steps is to generate the trajectory to follow intermediate writhe matrices from S_i and S_m, and the second steps is to generate the trajectory to follow intermediate writhe matrices from S_j and S_m. Then the first trajectory is connected with the reversal of the second trajectory to create a trajectory for *crossover* primitive. This is because the IK solver tends to fail to find a global optimum solution when the total writhe value of the writhe matrix is small.

Although there is not much change in skill parameters of minor fingers in each movement primitive after the refinement, it is still necessary to provide a method to create a sequence of intermediate writhe matrices for them. Changes in minor fingers are from peak-wm to peak-wm within the same sign of writhe. Consider $\mathcal{P}_i(w^i, p_x^i, p_y^i)$ and $\mathcal{P}_j(w^j, p_x^i, p_y^i)$ as skill parameters for initial peak-wm and final peak-wm writhe matrix. A sequence of intermediate writhe matrices can be created by first linearly interpolating between skill parameters of initial and final peak-wm to obtain skill parameters for all intermediate peak-wm. Then each intermediate writhe matrix is synthesised from its skill parameters.

Figure 6.8 illustrates a visualization of how the intermediate writhe matrices are interpolated for the major finger and Figure 6.9 for the minor fingers of all movement primitive. Note that linearly interpolation of skill parameters between two span-wm (with the same sign of writhe), that has been used in the process of creating intermediate writhe matrices, results in rotation of a span-line of the span-wm.

6.3 Connecting Movement Primitives

A sequence of movement primitives is passed to the robot system to be mapped onto the robotic hand. A trajectory of joints/hand for each movement primitive are generated separately. This section explains how to connect them together into a regrasping movement that is resembled to the one observed from human. A method to connect two movement primitives can be divided into two cases: a.) a case when a final grasping posture of the former movement primitive is the same as an initial grasping posture of the latter movement, or in other word, they possess the same skill parameters obtained from human, b.) a case when a final grasping posture of the former movement primitive is not the same as an initial grasping posture of the latter movement primitive, but they are in the same state in the taxonomy.

In the former case, connecting two consecutive movement primitives is simple. Since they are originally connected through the grasping posture (a final posture of the former and a final of the latter), both trajectories can directly be attached to each other. Note that in this case, the orientation of the pen obtained from the demonstration for both movement primitives are always the same, due to the fact that both movement primitives are connected during the demonstration and the initial assumption that an orientation of the pen does not change within the same movement primitive.

Based on a method how a range of each movement primitive is identified explained in Section 5.2.2.1, the latter case can only occur at the state which all three writhe matrices are peak-wms, i.e. all states with $t_p i_p m_p$. In other word, this case can only occur when the former movement primitive is *attaching* and the latter movement primitive is *detaching*. Note the method to connect these two movement primitives can also be used to connect the initial grasping posture of the regrasping movement to the first *detaching* movement primitive, or to connect the last *attaching* movement primitive to the final grasping posture of the regrasping movement.

In the latter case, if the orientation of the pen of both trajectories are the same, an interpolation in the topological space can be used to connect the final grasping posture of the former movement primitive to the initial grasping posture of the latter. Both grasping postures are described by skill parameters of three peak-wms, the trajectory to connect them can be interpolated by the method

explained in Section 6.2. By applying a method formerly used to generate intermediate writhe matrices of the minor fingers of the movement primitive to all fingers, the intermediate writhe matrices that connects between the two grasping postures can be created; thus the hand/joint trajectory to connect them.

However, if the orientation of the pen changes, this can be very complicated. When a robotic hand tries to move or change an orientation of a grasped object without any contact between fingers and the object being detached, the movement can be referred to as *rolling* or *sliding* [Bicchi 2000]. In this system, this movement is created manually by the human operator by considering appropriate trajectories for the fingertips of all fingers and using a simple IK solver to generate a hand/joint trajectory.

6.4 Experimental Results

Sequences of human regrasping movement considered in Section 4.4 and Section 5.3 are mapped to the robotic hand described in Appendix B. Regrasping movement is recognised into a sequence of movement primitives and their skill parameters. These skill parameters are refined before they are used to generate a movement. Trajectories of hand and finger movement are generated for all movement primitives in the sequence and connected together to create a regrasping movement for a robotic hand that imitates the observed human movement.

6.4.1 Results of Skill Refinement

Figure 6.10 and Figure 6.11 illustrate refinement results for skill parameters obtained from Interdigital Step – Sequences 1 and 2 respectively. Both original and refined skill parameters are shown for comparison. The skill parameter shown is a p_y of peak-wm, which represents a contact location on the pen of the corresponding finger. In each small image, contact locations of all three fingers are shown, except when the corresponding writhe matrix is span-wm, a l_y is displayed instead.

It is noticeable that within a movement primitive, contact locations of both minor fingers are refined to be unchanged from the initial state to the final state. As a result, for all consecutive movement primitives, the contact locations of their minor fingers are unchanged throughout, e.g. contact location of thumb (red) and middle finger (blue) are unchanged from frame 22^{nd} to 87^{th} in the refined version of Figure 6.10.

Regarding the threshold values used in refining constraints, they are assigned empirically excepts for **Constraint 3**. For **Constraint 1** and **Constraint 2**, $\delta_1, \delta_2, \delta_3$ are assigned equally to 0.1 throughout the experiment. This is equal to $^{30}/_{2} * 0.1 = 1.5$ centimetres, when a 30 centimetres pen is used. For **Constraint**

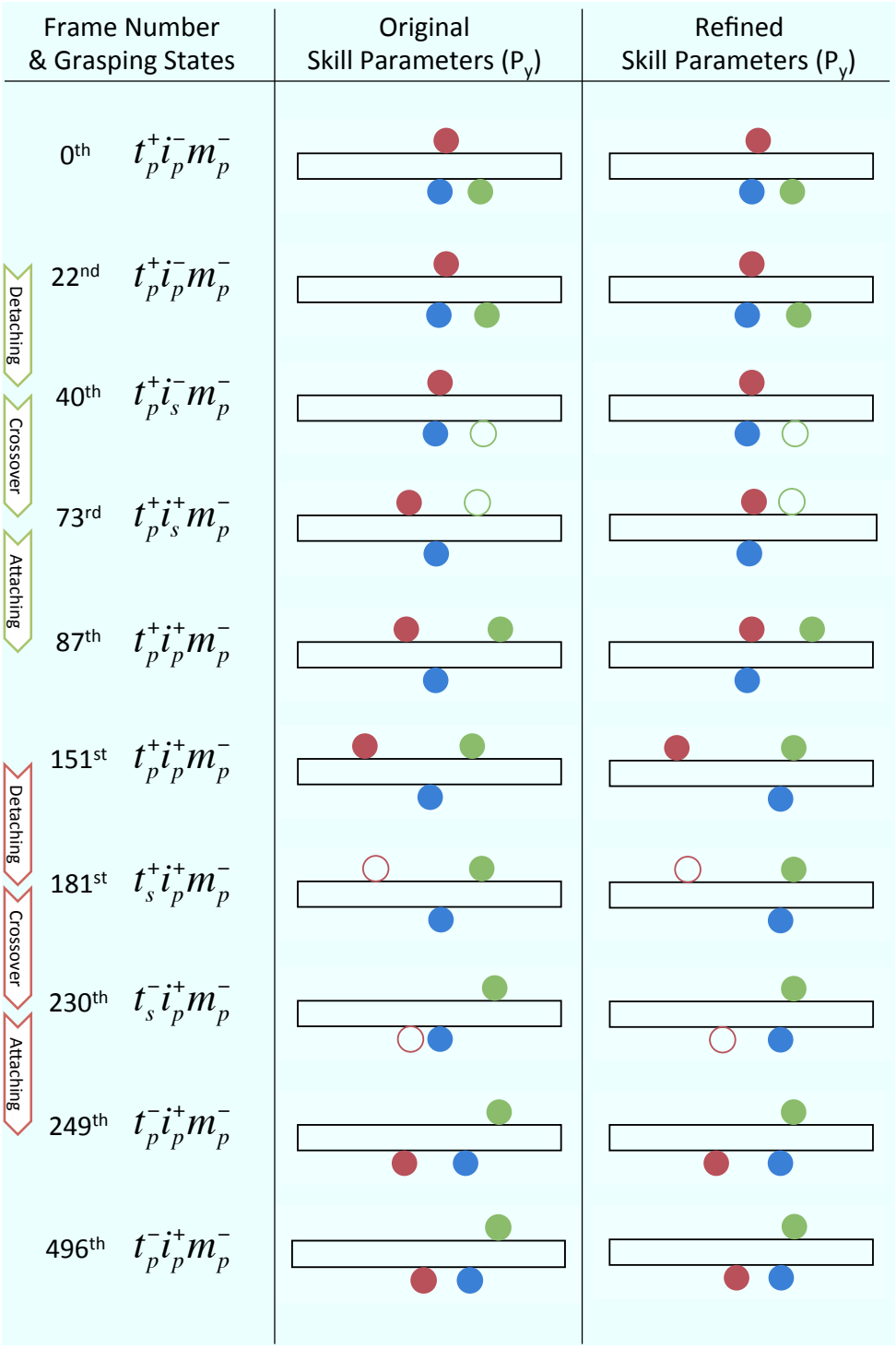


Figure 6.10: Refinement of the skill parameters obtained human demonstration of Interdigital Step – Sequences 1. The left column shows original skill parameters, where the right column shows the refined version. For a peak-wm (a filled circle), a contact location on the pen p_y is shown, where for a span-wm (an unfilled circle) an average tangled location on the pen l_y is shown instead (red=thumb, green=index finger, blue=middle finger).

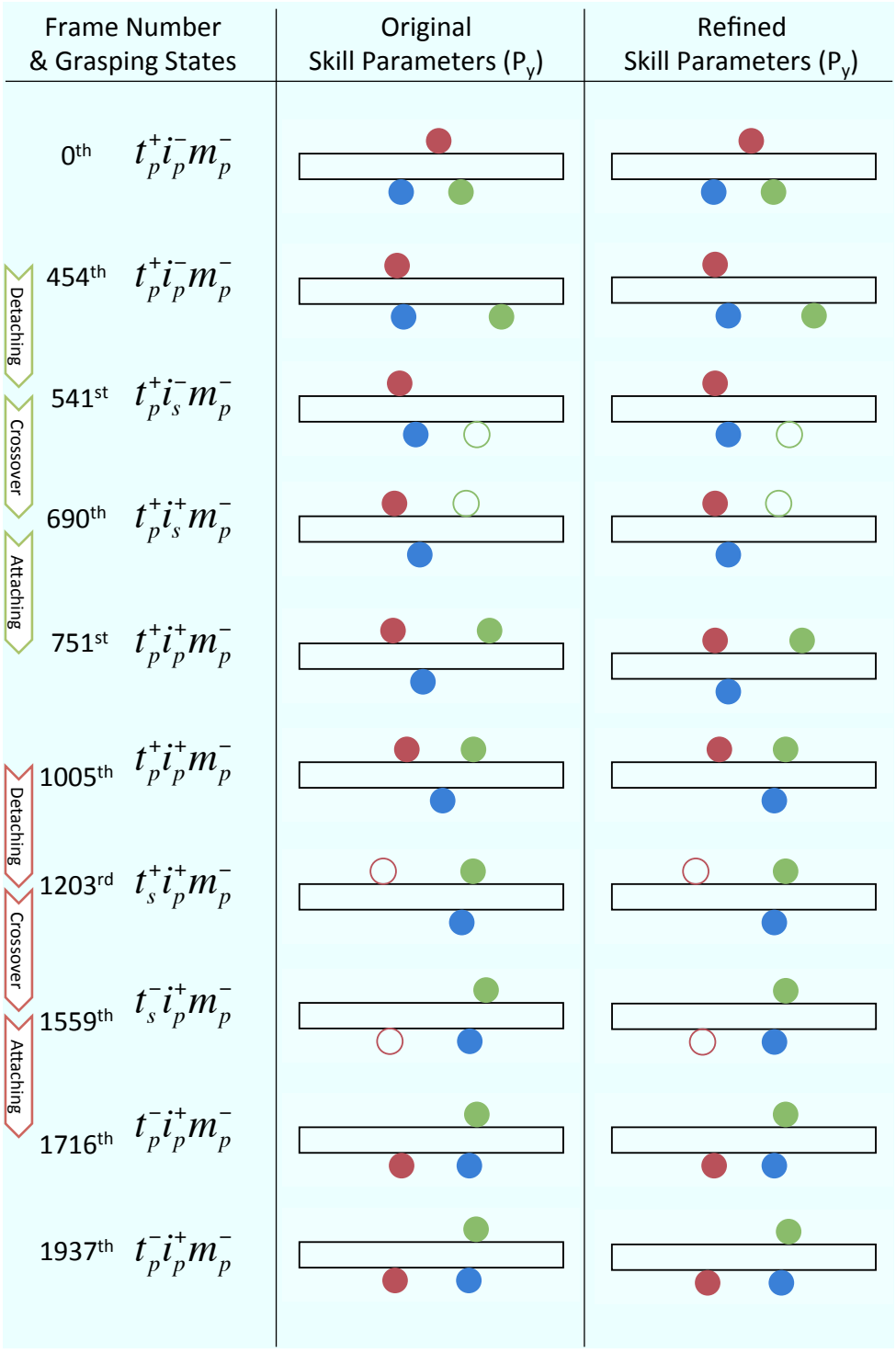
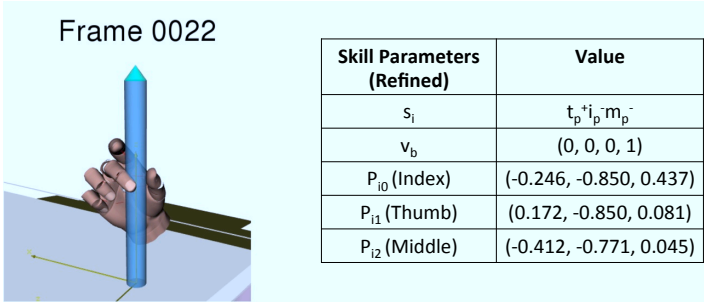
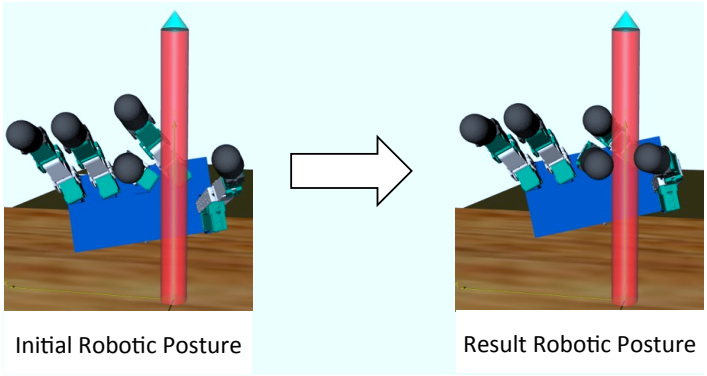


Figure 6.11: Refinement of the skill parameters obtained human demonstration of Interdigital Step – Sequences 2. The left column shows original skill parameters, where the right column shows the refined version. For a peak-wm (a filled circle), a contact location on the pen p_y is shown, where for a span-wm (an unfilled circle) an average tangled location on the pen l_y is shown instead (red=thumb, green=index finger, blue=middle finger).



Skill Parameters (Refined)	Value
s_i	$t_p^+ i_p^- m_p^-$
v_b	(0, 0, 0, 1)
P_{i0} (Index)	(-0.246, -0.850, 0.437)
P_{i1} (Thumb)	(0.172, -0.850, 0.081)
P_{i2} (Middle)	(-0.412, -0.771, 0.045)

(a) a human grasping posture (frame 22nd) and its skill parameters.



(b) Initial robotic posture and the result robotic posture of the mapping.

Figure 6.12: Skill parameters obtained from human is mapped to a robotic hand (frame 22nd of Interdigital Step – Sequence 1).

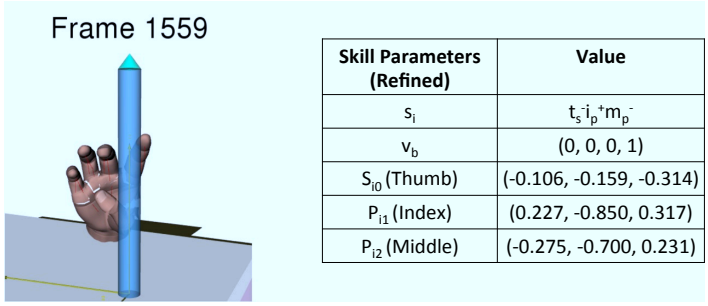
3, a feasible contact area on the fingers are measured and $l_{min} = -0.85, l_{max} = -0.70$ are selected.

6.4.2 Results of Movement Mapping

To map a human regrasping movement onto the robotic hand, trajectories of all movement primitives are generated and connected together. In order to generate a trajectory for each movement primitive, initial and final grasping postures of the robotic hand are first created from the corresponding skill parameters. Then a trajectory that connected between these two grasping postures is generated.

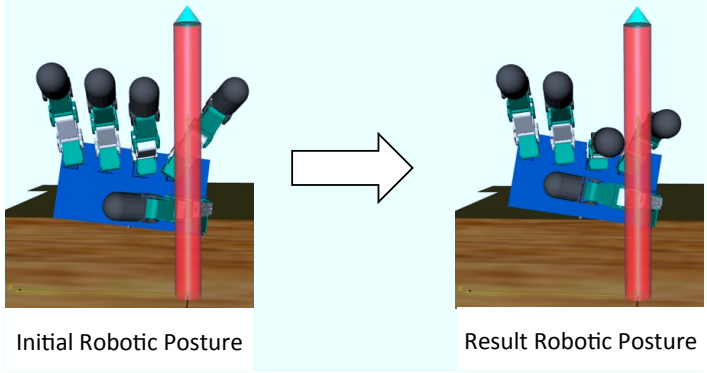
6.4.2.1 Mapping Skill Parameters to Robot Postures

Figure 6.12 illustrates a robotic posture that is mapped to skill parameters that obtained from human demonstration (frame 22nd of Interdigital Step – Sequence 1). Three writhe matrices, all peak-wms in this case, are synthesised from their



Skill Parameters (Refined)	Value
s_i	$t_s^- i_p^+ m_p^-$
v_b	(0, 0, 0, 1)
S_{i0} (Thumb)	(-0.106, -0.159, -0.314)
P_{i1} (Index)	(0.227, -0.850, 0.317)
P_{i2} (Middle)	(-0.275, -0.700, 0.231)

(a) a human grasping posture (frame 1559th) and its skill parameters.



(b) Initial robotic posture and the result robotic posture of the mapping.

Figure 6.13: Skill parameters obtained from human is mapped to a robotic hand (frame 1559th of Interdigital Step – Sequence 2).

corresponding skill parameters using method explained in Section 6.2.1.1. Then, they are used as a target tangle relation for solving an IK problem. The robotic posture shown in the left of Figure 6.12(b) is used as a initialising hand configuration. The IK solver then iteratively searches for a solution, until the criteria is met which yields a resulting posture on the right of the figure. Note that the criteria used for a stop condition is $\|\Delta \mathbf{r}\|$, where $\Delta \mathbf{r}$ is a small change in the generalized coordinate of the robotic hand.

Similarly, Figure 6.13 displays a robotic posture that is mapped to skill parameters that obtained from human demonstration (frame 1559th of Interdigital Step – Sequence 2). However, the grasping posture contains a finger (thumb) whose writhe matrix is a span-wm. Thus, a method to synthesise a target tangle relation is slightly different. In addition, the initialising hand configuration for the IK problem, shown in the left of Figure 6.13(b), is also different due to the state of human posture (i.e. $t^- i^+ m^-$).

As mentioned in Section 6.2.1.1, the value of σ_x and σ_y, when peak-wm is synthesised, effect a distance between strands of the finger and the pen. The

distance indicates whether the finger is in contact with the pen or not. In the current implementation, these value are empirically and manually assigned so that the finger is in contact with the pen. The range of assigned values are range between 0.6 to 1.5 depending on obtained writhe value and the length of a target robotic finger. This process can be automated by using a collision detection engine to identify the value that would lead the finger to be in contact with the pen.

In addition, in a current implementation, the setting of σ_x and σ_y are also used to solve the problem of an inaccuracy in the data acquisition phase. For instance, it can be seen clearly in frame 40^{th} of Figure 5.12 that the thumb is not in contact with the pen, but its writhe matrix is classified as a peak-wm. This results in the low value of total writhe matrix of the peak-wm that is passed on to the robot as a skill parameter; thus lower values of σ_x and σ_y used in the synthesized process.

An alternative approach to tackle the issue is to find a specific writhe value the is required for each finger of a particular robotic hand to be in contact with the pen, and set it to all corresponding peak-wms during skill refinement. This would narrow down the range of possible value to be assigned to σ_x and σ_y. However, this approach should not be used to refine a span-wm as its writhe value also reflects the average distance from the finger to the pen.

6.4.2.2 Mapping of Regrasping Movement

Figure 6.14 and Figure 6.15 show results of a robot imitating human demonstration of Interdigital Step – Sequence 1 and 2 sequentially. Trajectories of all movement primitives are generated and connected together. The connected trajectory is passed onto the robot to be executed. In current implementation, the robot uses no force or visual feedback during an execution of the trajectory.

It is noticeable from the results that the pen orientation changes during the imitation. However, in the demonstration of these two sequences, Interdigital Step – Sequence 1 and 2, the orientation of the pen does not change at all, so as to the result from the mapping. A small improvement on this issue can be achieved by tuning the opposition distance threshold using during the refinement or the assigned values of σ_x and σ_y using during the synthesised of peak-wms. An alternative improvement would be to incorporate force and visual feedback into the robot system, and an algorithm to real-timely modify the hand trajectory. However, the alternation in the mapping framework could be expected in this case.

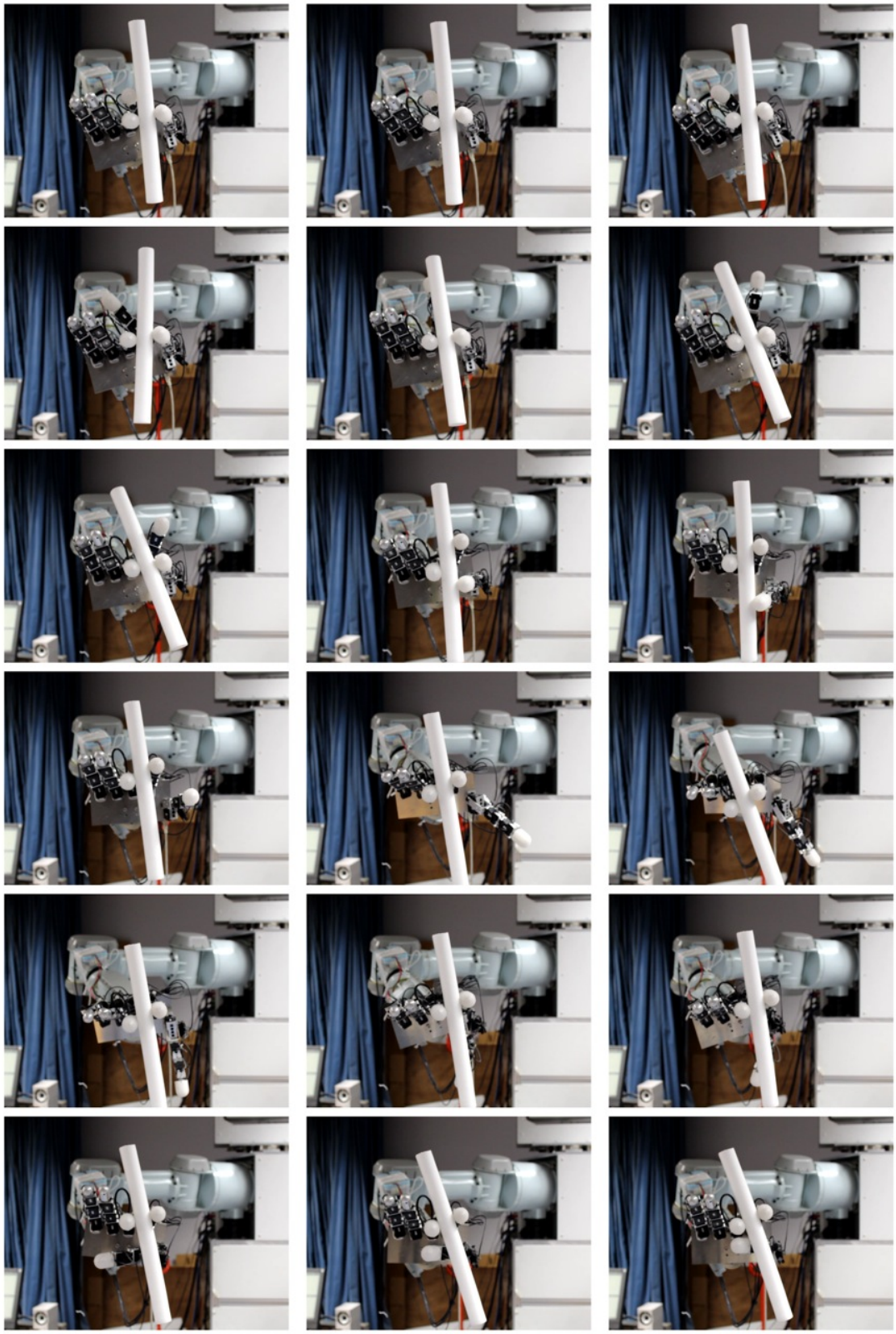


Figure 6.14: A result of the robotic hand imitates a human regrasping movement (Interdigital Step – Sequence 1).

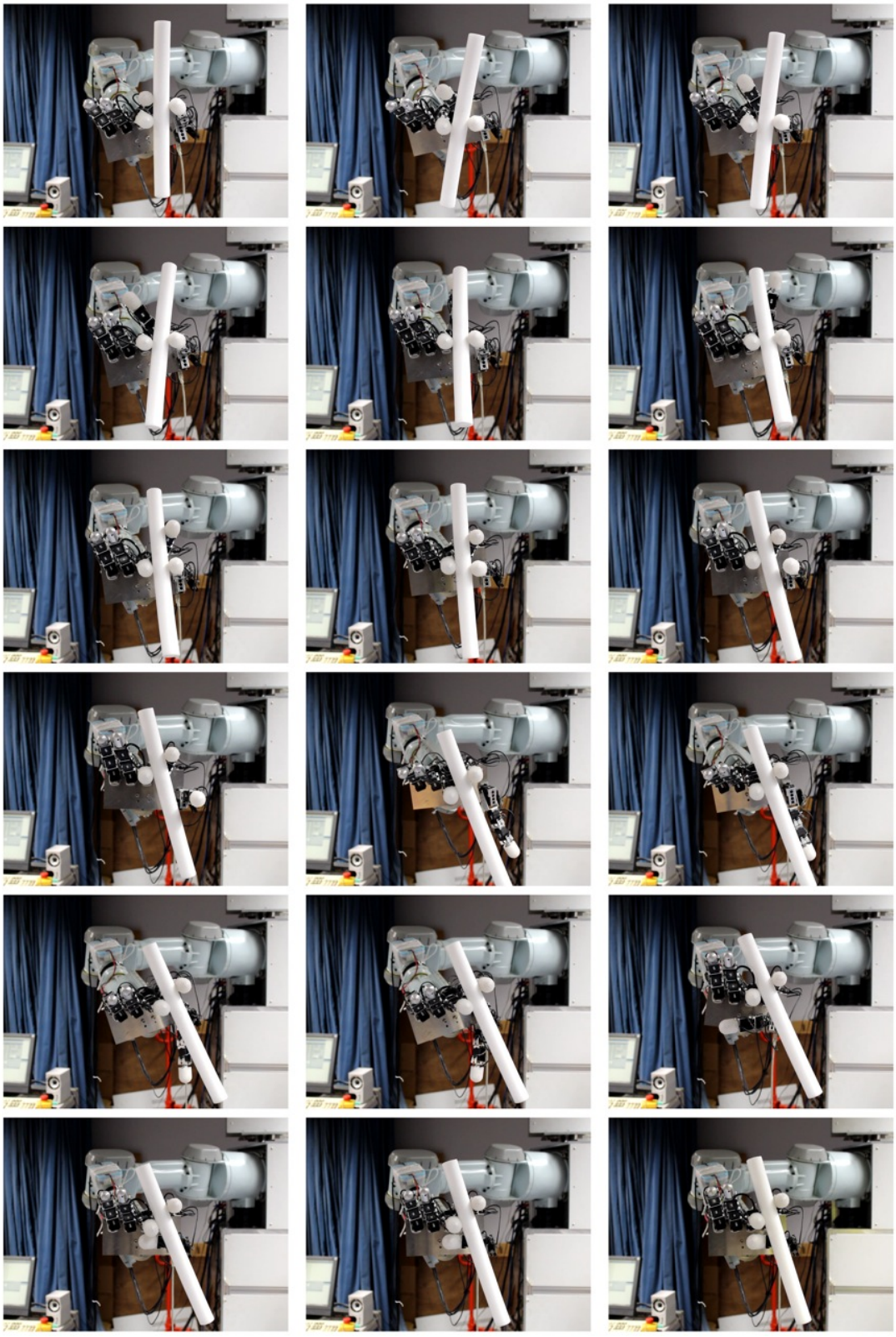


Figure 6.15: A result of the robotic hand imitates a human regrasping movement (Interdigital Step – Sequence 2.

6.5 Summary of Chapter

A framework to map human regrasping movement to a robotic hand is described. Human regrasping movement is represented as a sequence of movement primitives. A trajectory for each movement primitive is generated based on their skill parameters and connected together to create the regrasping movement for a robotic hand. The trajectory of each movement primitive is generated by interpolating the hand in a topological space. Based on the skill parameters of initial and final writhe matrices of each movement primitive, the interpolation method for all three movement primitives are proposed. It creates intermediate writhe matrices which, in turn, are used to lead the interpolation of the robotic hand in a topological space.

Skill parameters obtained from human demonstration cannot be used directly to generate a trajectory for a robotic hand that can be successfully executed in the real world. A method to refine these skill parameters is proposed. Finally, a regrasping movement for a robotic hand is generated using the proposed framework and these refined parameters and is successfully executed on a robotic hand in the real world.

CHAPTER

SEVEN

Conclusions

Contents

7.1 RPO – A Summary

This thesis describes a novel approach to teach a regrasping movement to a robot. The approach is based on a LFO paradigm, which allows the robot to learn how to perform a certain task by observing and imitating human. Robot plans its own regrasping movement by observing human demonstrating a movement, segmenting and recognising the movement based on a pre-defined task model. This allows an observed regrasping movement to be represented by a sequence of movement primitives. The robot then imitates the movement by sequentially executing these movement primitives using skill parameters obtained in the planning process.

The thesis focuses on the changes of topological relation, referred to as tangle topology, between a hand and an object in a regrasping movement. Hands and a manipulated object are substituted with a set of zero-width strands. Tangle relation examines a relation between these strands. Features based on this relation are used to temporally segment a captured regrasping movement into various meaningful segments. By comparing segmented results, the segmentation of based on a change in types of the feature called writhe matrices is chosen to build a task model upon.

A task model is consisted of two components: movement primitives and skill parameters. In the RPO system, movement primitives are defined based on

changes in types of writhe matrices between fingers and the object. A set of skill parameters for each type of movement primitive are a necessary information needed to transfer the movement primitive from a human hand to a robotic hand. Both movement primitives and skill parameters are recognised and extracted from human demonstration, which turns a captured regrasping movement into a strategy or a plan for a robot to execute it.

A sequence of movement primitives together with their skill parameters are used by a robot to imitate the captured regrasping movement. Skill parameters of all movement primitives are refined to comply with physical constraints of a target robotic hand and environment before the movement primitives are sequentially executed. A mapping framework to map each movement primitive to a robotic hand is described. It is based on the interpolation of a robotic hand in a topological space and composed of two steps: 1.) mapping skill parameters to create an initial and a final postures of a robotic hand and, 2.) creating a trajectory of the hand to connect between the two postures.

Once trajectories for each movement primitives are created, they are connected together to create a regrasping movement for a robotic hand. A robot imitates the movement by simply following the pre-constructed trajectory. A robot system used to execute the movement comprised of a 20 DOFs custom-made robotic hand, attached to the Mitsubishi PA-10 robotic arm to maneuver around. The hand is build to resemble the DOF structure of the human hand.

7.1.1 Discussion

The approach described in this thesis has various advantages over other methodologies. Firstly, RPO system is based on a LFO paradigm. It can be seen that the movement of the robotic hand produced from the proposed system is similar to that of a human. By mimicking the way a human regrasps an object, the generated movement will in include knowledge of various constraints of the real world environment. These include object constraints and task constraints. By comparing to automatic programming approaches, many of these constraints are ignores which may make the approach struggle when applied outside a controlled environment [Cutkosky 1989].

Secondly, once human performs a certain regrasping movement, the knowledge recognised and extracted can be used to create a movement on many hands. Due to the limitation of resources, the method has not been tested with other types of robotic hand. However, it is believed that if their structure are not too different from that of the human hand, the proposed method should work well with them with some parameter tuning.

Thirdly, an observation process requires poses and configuration of a human hand and an object at every frame in the sequence. In this thesis, a *CyberGlove* data glove and a *Polhemus* motion tracker are used to capture these data. How-

ever, this can be replaced by other devices (e.g. vision-based) that provide a better accuracy. Although a force/tactile information might improve the segmentation method that separates between two types of writhe matrix, there is no need for the use of a tactile sensor. This is because contact relations during regrasping are indirectly inferred from a topological representation, which only requires information of poses.

Lastly, a topological information chosen as a main representation in the system contains more information than a set of contact locations (the object) which generally uses to represented in a planning of regrasping movement. A peak-wm contains contact locations both on the object and on the hand. A span-wm stores information of a grasping posture when there is no contact between the hand and the object. A combination of both types of writhe matrix in a sequence, not only provides an information where the contacts should be, it also includes an information of the trajectory of the fingers (regards to the object) during regrasping. This allows a trajectory of a robotic hand to be easily interpolated in a pre-defined path in a topological space.

7.2 Future Directions

Although the proposed RPO system allows a robot to imitate a regrasping movement from human demonstration, it has reduced and omitted various conditions and constraints. The following directions can be done to improve the capabilities of the system.

The current system only considers a pen-like object, or an object that is can be represented by one straight strand. It would be interesting to explore into the regrasping movement of other objects. To begin with, if the object is represented by a strand that is not straight, tangle relation might not remains the same during regrasping. Furthermore, if regrasping movements of more complicated objects that cannot be represented by one strand are considered, the foremost question would be how to represent them as strands.

The current taxonomy of grasping postures considers a tangle relation of three fingers. This results in the movement primitives that allow only one finger to greatly move (a major finger) because other fingers (minor fingers) need to keep the stability of the object. If a taxonomy is extended to more fingers or to other components of the hand (e.g. palm), it would allow a regrasping movement that contains a simultaneous movement of more than one (major) finger to be imitated. However, this might also result in the modification of task model and the mapping framework.

A contact location provided by peak-wm as the strands do not possess a directional information on their side. In other word, for a particular grasping posture, if the pen rotates around its axis, the recognised contact information will remain the same. Including this information in the recognition process would

enhance a knowledge obtained from human demonstration and provides more precious imitation on the robot. By doing this, a method that is used to refine skill parameters before mapping to the robotic hand would need to be enhanced to a three dimensional problem.

In current system, the object is not allowed to move when each movement primitive is performed. This is due to the lack of knowledge on how the tangle relation (i.e. writhe matrix) would be altered, if the movement primitive of a finger effects the location and orientation of the object. It is not immediately obvious what would be necessary to enabling this. However, at a very least, a change of tangle relation, due to a movement of a finger that moves the object, must be calculated.

Lastly, in current system a trajectory of a regrasping movement for a robotic hand is pre-calculated and simply playback on the robot. This dues to the lack of sensors on a robot system, e.g. force tactile sensors, visual sensors. It would be interesting to develop a feedback system on the robot when imitating a regrasping movement. A sequence of movement primitives can be executed one at the time, and examine its state before proceeding to execute the next one.

APPENDIX

A

Data Acquisition System

Contents

Main objective of developing data acquisition system is to let a robot observe a human regrasping movement. In other words, it is used for human to demonstrate a regrasping movement to a robot system. In this chapter, a detail information about the data acquisition system is explained. A possible improvement for the developed system are also discussed, together with other alternative system setups.

A.1 System Setup

At a particular moment, a data acquisition system should be able to capture a configuration (joint angles) of a human hand, and poses (position and orientation) of both a human hand and a manipulated object. These data are referred to as grasping posture. A regrasping movement are captured as a sequence of grasping postures concatenated together into a trajectory of grasping postures. In the system, a pen-like object (e.g. pen, paintbrush) is considered. The exact geometry of the manipulated object is known prior to the demonstration.

An obvious approach for the problem would be to use cameras to track the hand and object in real-time. However, this approach usually suffers from either an occlusion problem, or a captured rate due to a calculation of complicated tracking algorithm. A different approach would be to use a marker based tracking system [Chang *et al.* 2007]. However, after some experiments with the regrasping movement of interest, it seems to be suffering from the occlusion problem as well.

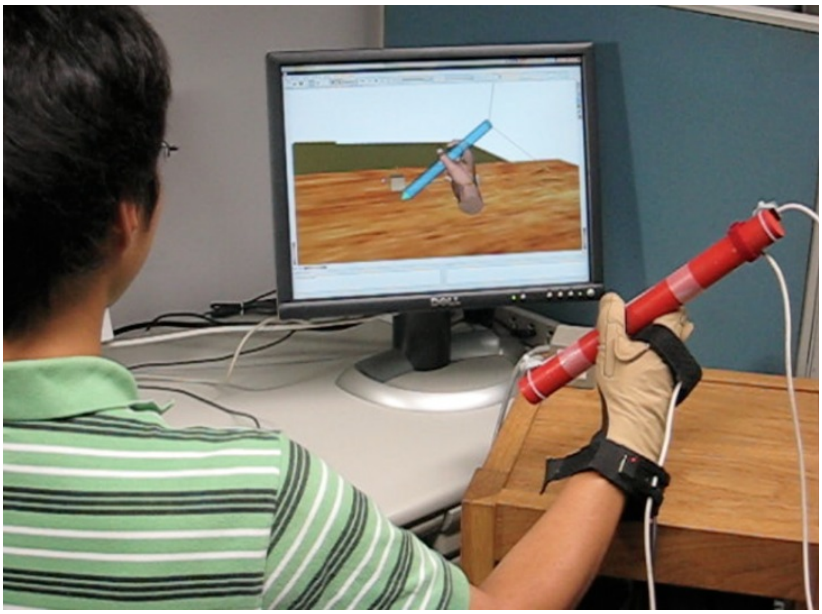
In this system, a *CyberGlove* data glove and *Polhemus* motion tracker are used to capture a demonstration of regrasping movement. Information from the glove and motion tracker could have been used directly. However, due to its lack of accuracy to capture hand configuration in the real world, data obtained from these sensors are connected to a virtual environment [Miller & Allen 2004] and the information in a virtual environment is used instead. In other word, a human demonstration of a regrasping movement is conducted in a virtual environment, while the glove and the motion tracker are used to interface with the models in a virtual environment. Figure A.1(a) illustrates how human interfaces with a virtual environment via the glove and the motion tracker.

A.1.1 Mapping Sensor Data to a Virtual Environment

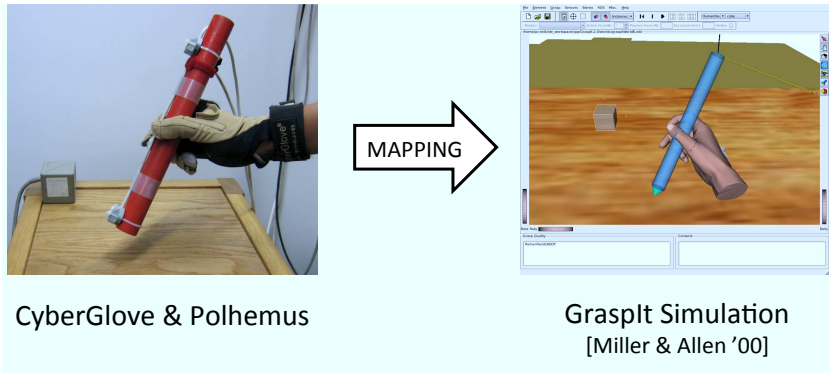
Data acquired from sensors are hand and object location from Polhemus motion tracker, and hand configuration from Cyberglove. The hand and object location are transformed and mapped directly into the virtual environment. As for a hand configuration, the data obtained from the sensors are mapped to the model according to their labels shown in Figure A.2. For $(h, c) \in \{(0, 8), (1, 4), (2, 5), (3, 11), (4, 6), (5, 7), (6, 11), (7, 9), (8, 10), (9, 14), (10, 12), (11, 13), (12, 0), (13, 3), (14, 1), (15, 2)\}$,

$$H[h] = G_{h,c}C[c], \tag{A.1}$$

where $G_{h,c}$ is a gain parameter different for each (h, c) pair. These gain parameters are identified manually based on trail-and-error. In general, accuracy of the values are acceptable, except abduction joints and the thumb that usually required a fine tuning for different grasping postures. Alternative approaches to calibrate these variables or mapping a glove to a particular hand model could be used to increase an accuracy [Fischer *et al.* 1998, Griffin *et al.* 2000, Maycock *et al.* 2011]. Though, they are usually applicable to increase an accuracy for only some grasping postures or movement.

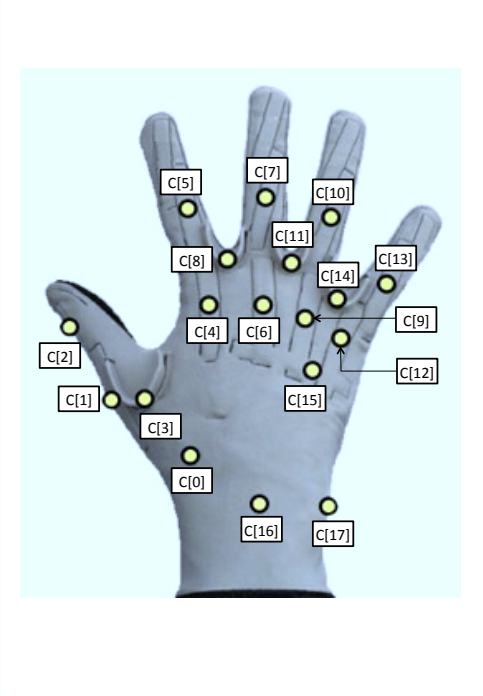


(a) Human demonstrates a movement through a data acquisition system.

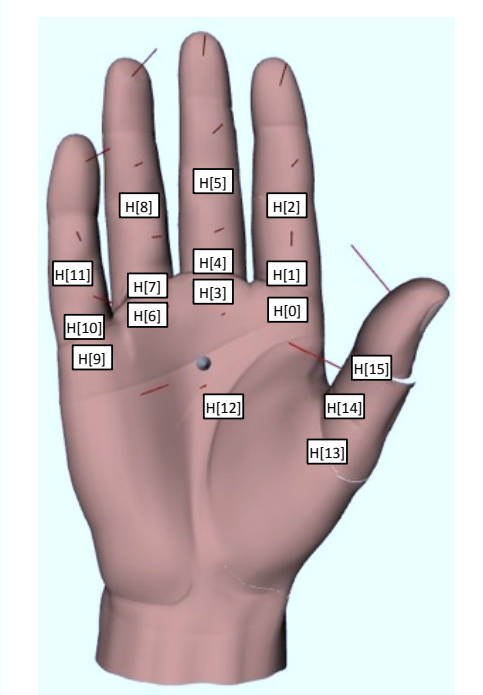


(b) Data captured from Cyberglove and Polhemus are mapped to a virtual environment [Miller & Allen 2004].

Figure A.1: A data acquisition system.



(a) The 18-sensor Cyberglove.



(b) Human hand model (16 degree of freedoms) provided in GraspIt simulation.

Figure A.2: Joint numberings of Cyberglove sensor and human hand model.

APPENDIX

B

Robot System

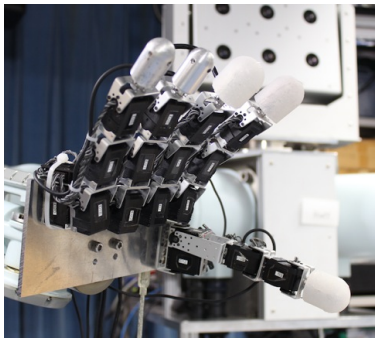
Contents

A robot system used to imitate human regrasping movement is described in this chapter. A robotic hand is made to resemble a human hand configuration. It consists of 20 degrees of freedom and attached to Mitsubishi PA-10 robotic arm to be maneuvered around.

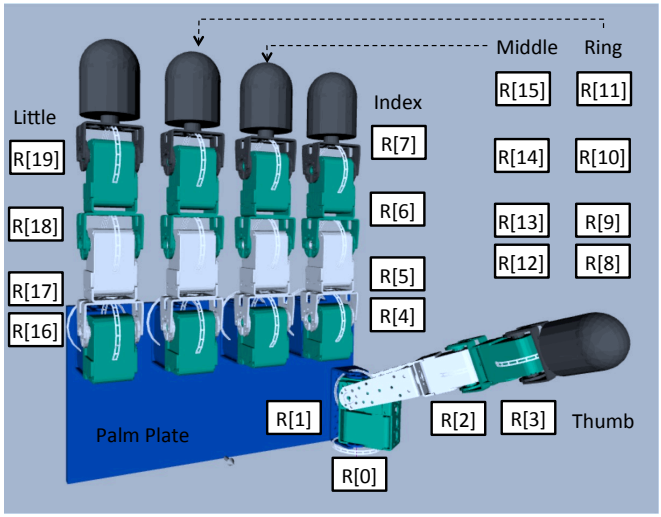
B.1 Configurations of a Robotic Hand.

A custom-made robotic hand is constructed by attached 20 Futaba RS303MR servo motors together as shown in Figure B.1. It consists of five fingers, named similar to fingers of human hand. Each finger contains four degrees of freedom. All fingers, excepts the thumb, have the same kinematic configurations. Denavit-Hartenberg parameters (DH parameters) of the thumb and other fingers is given in B.1.

All five fingers are attached to a rectangular aluminium palm plate. Dimensions of the palm plate are $200 \times 100 \times 5$ millimeters Consider an origin to be at the mid-bottom of the front of the plate as shown in Figure B.1(b), the transformation from the origin of the palm plate to the finger bases are shown in Table B.2.



(a) A robotic hand attached with a Mitsubishi PA-10 robotic arm.



(b) Joints information of the robotic hand.

Figure B.1: A custom-made 20 degrees of freedom robotic hand.

(a) DH parameters of the thumb.

θ (deg.)	d (mm.)	a (mm.)	α (deg.)
$R[0] - 90$	36.865	0	90
$R[1]$	0	80.350	90
$R[2]$	−0.116	47.481	0
$R[3]$	−0.818	63.061	0

(b) DH parameters of index finger (and the others).

θ (deg.)	d (mm.)	a (mm.)	α (deg.)
$R[4]$	29.364	1.019	90
$R[5]$	−0.127	47.481	0
$R[6]$	−0.016	47.482	0
$R[7]$	−0.140	63.061	0

Table B.1: Denavit-Hartenberg parameters of fingers of the robotic hand.

Finger	**Translation** (mm.)	**Orientation** (zyx in deg.)
Thumb	$(94.431, 1.545, 20.092)$	$(0, 0, -90)$
Index	$(52.650, 98.321, 15.998)$	$(90, 0, 0)$
Middle	$(5.150, 98.321, 15.998)$	$(90, 0, 0)$
Ring	$(-42.350, 98.321, 15.998)$	$(90, 0, 0)$
Little	$(-89.850, 98.321, 15.998)$	$(90, 0, 0)$

Table B.2: Transformation of each finger to a base of the robotic hand.

APPENDIX

C

Input Data

Contents

This chapter illustrates regrasping movements capture from a data acquisition system described in Appendix A. These movements are used as input data for the experiments conducted in this research.

C.1 Interdigital Step

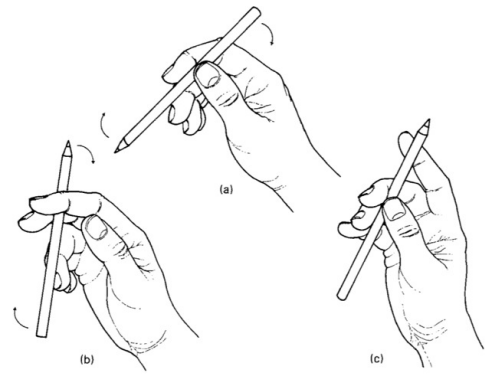


Figure C.1: Interdigital step movement.

Interdigital step is a manipulative movement that applied to long, thin objects [Elliott & K.J. 1984]. Three fingers move and combines to achieve the

movement as shown in Figure C.1. The reversal of this movement is captured and shown in the following subsections.

C.1.1 Interdigital Step – Sequence 1

The captured movement consists of 497 frames. It is divided and illustrated sequentially in Figure C.2, Figure C.3, and Figure C.4.

C.1.2 Interdigital Step – Sequence 2

The captured movement consists of 1938 frames. It contained a lot more frames than one in Appendix C.1.1 due to the higher frame rate of the data acquisition system. It is divided and illustrated sequentially, one every four frames, in Figure C.5, Figure C.6, and Figure C.7.

Figure C.2: Interdigital Step – Sequence 1 : Frame 0^{th}-179^{th}.

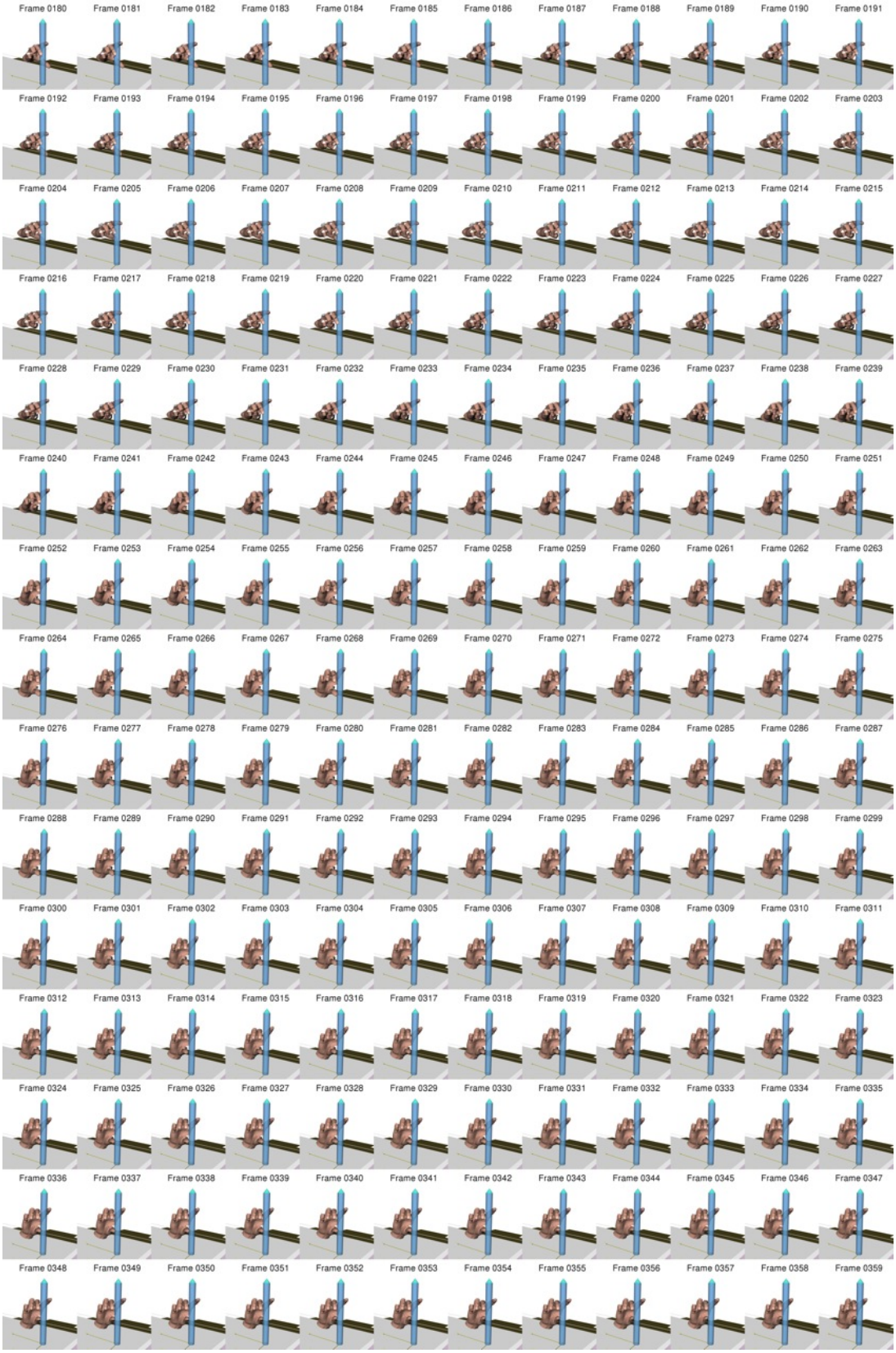


Figure C.3: Interdigital Step – Sequence 1 : Frame 180^{th}-359^{th}.

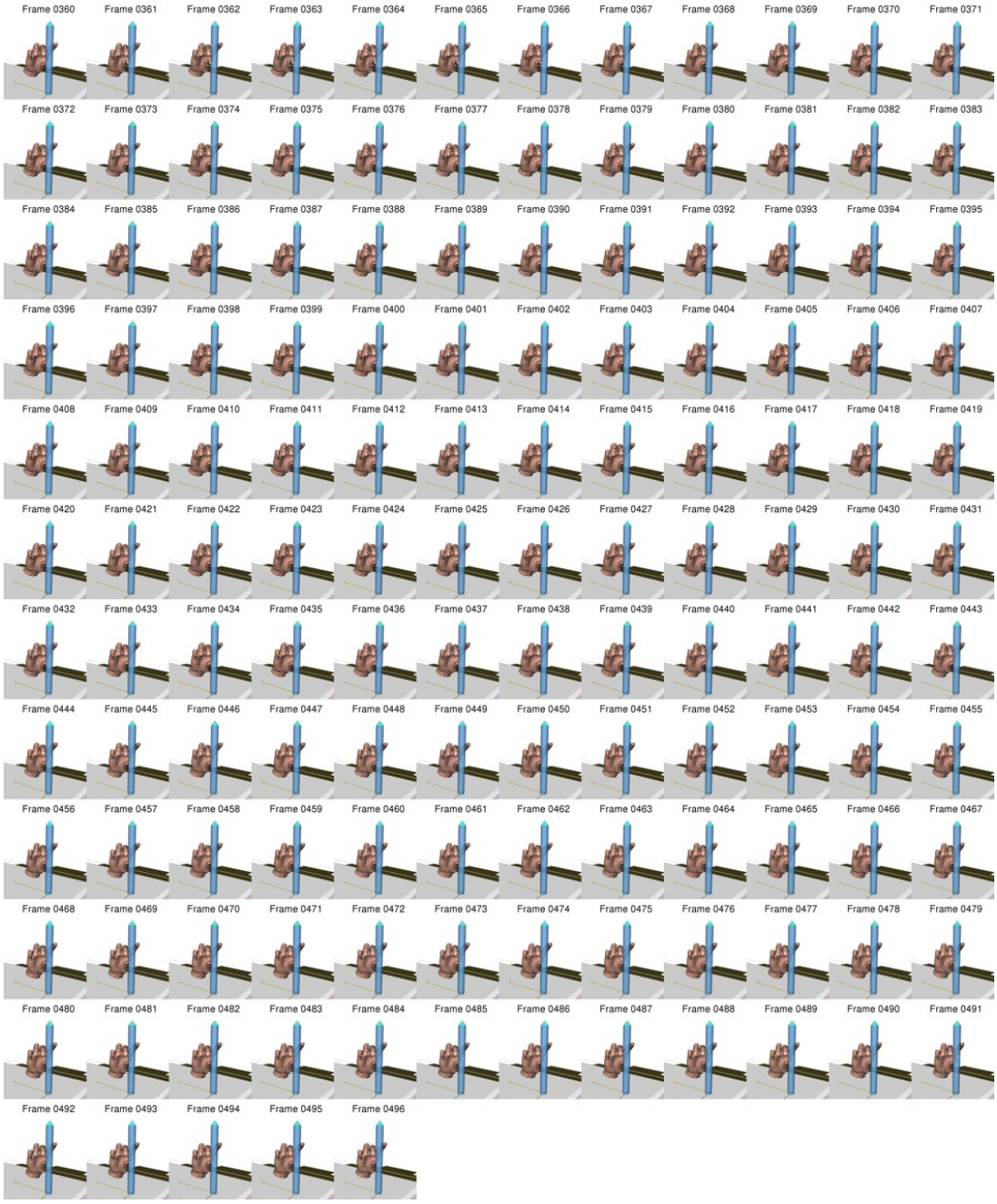


Figure C.4: Interdigital Step – Sequence 1 : Frame 360$^{\text{th}}$-496$^{\text{th}}$.

Figure C.5: Interdigital Step – Sequence 2 : Frame 0^{th}-716^{th} (one every four frames).

Figure C.6: Interdigital Step – Sequence 2 : Frame 720^{th}-1436^{th} (one every four frames).

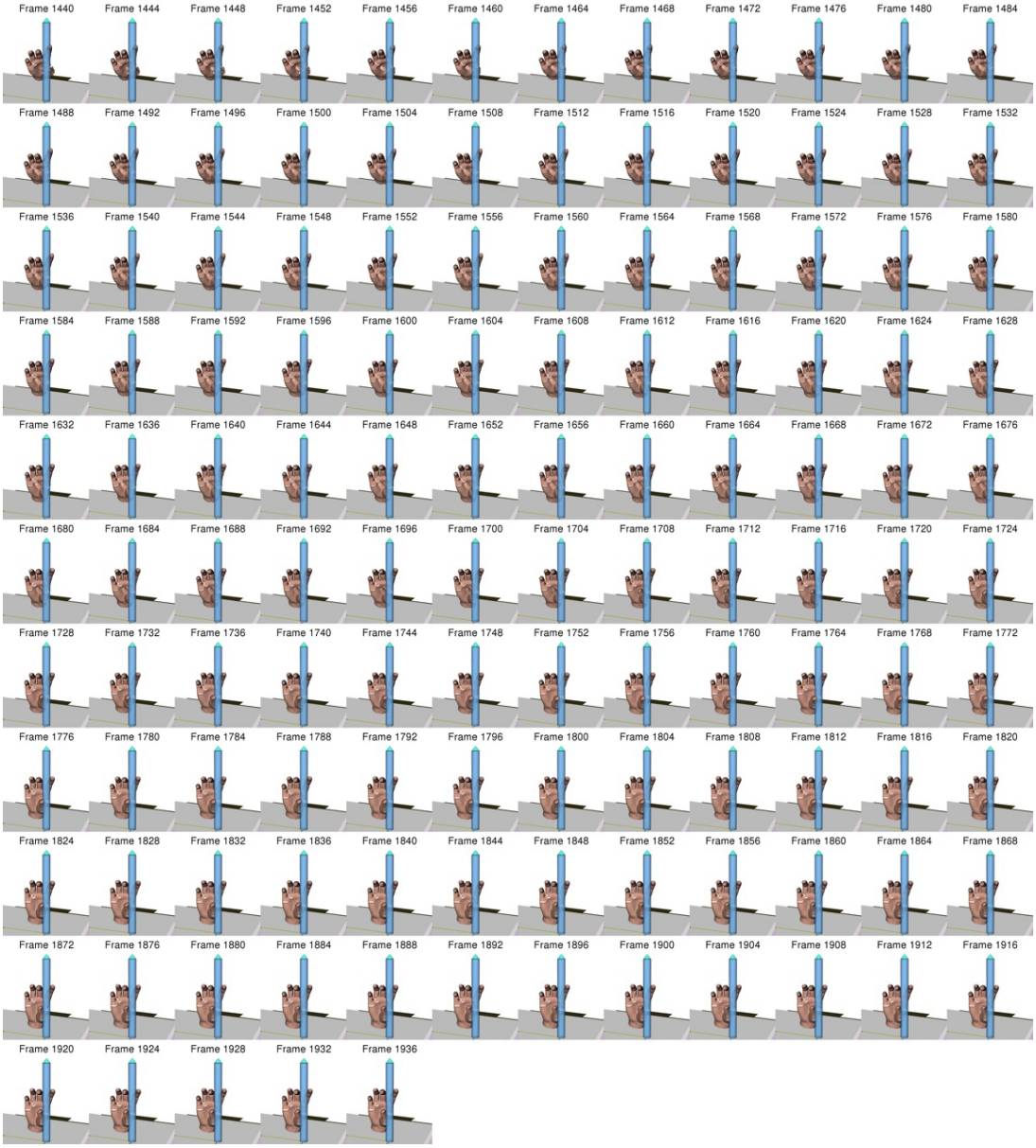


Figure C.7: Interdigital Step – Sequence 2 : Frame 1440$^{\text{th}}$-1936$^{\text{th}}$ (one every four frames).

BIBLIOGRAPHY

[Aleotti & Caselli 2006] J. Aleotti and S. Caselli. *Grasp recognition in virtual reality for robot pregrasp planning by demonstration.* In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, pages 2801 –2806, may 2006.

[Aleotti & Caselli 2007] J. Aleotti and S. Caselli. *Robot grasp synthesis from virtual demonstration and topology-preserving environment reconstruction.* In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pages 2692 –2697, 29 2007-nov. 2 2007.

[Ambrose *et al.* 2000] R.O. Ambrose, H. Aldridge, R.S. Askew, R.R. Burridge, W. Bluethmann, M. Diftler, C. Lovchik, D. Magruder and F. Rehnmark. *Robonaut: NASA's space humanoid.* vol. 15, no. 4, pages 57 –63, jul/aug 2000.

[Arbib *et al.* 1985] M.A. Arbib, T. Iberall and D.M. Lyons. *Coordinated control programs for movements for the hand.* In W. Goodwin and Darian-Smith, editors, Hand Function and The Neocortex, pages 111–129. Springer-Verlag, Berlin, 1985.

[Au & Woo 2004] Chikit K. Au and Tony C. Woo. *Ribbons: Their Geometry and Topology.* Computer-aided Design and Applications, vol. 1, pages 1–6, 2004.

[Au 2008] Chikit K. Au. *The Geometric Interpretation of Linking Number, Writhe and Twist for a Ribbon.* Computer Modeling in Engineering and Sciences (CMES), vol. 29, no. 3, pages 151–162, 2008.

[Bann *et al.* 2003] Simon Bann, Mansoor Khan, Juan Hernandez, Yaron Munz, Krishna Moorthy, Vivek Datta, Timothy Rockall and Ara Darzi. *Robotics in surgery.* Journal of the American College of Surgeons, vol. 196, no. 5, pages 784 – 795, 2003.

[Bekey *et al.* 1990] George A. Bekey, Rajko Tomovic and Ilija Zeljkovic. *Control architecture for the Belgrade/USC hand.* In S. T. Venkataraman and T. Iberall, editors, Dextrous Robot Hands, pages 136–149. Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[Bernardin *et al.* 2005] K. Bernardin, K. Ogawara, K. Ikeuchi and R. Dillmann. *A sensor fusion approach for recognizing continuous human grasping sequences using hidden Markov models.* IEEE Transactions on Robotics, vol. 21, no. 1, pages 47–57, Feb. 2005.

[Bicchi 2000] A. Bicchi. *Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity.* IEEE Transactions on Robotics and Automation, vol. 16, no. 6, pages 652 –662, dec 2000.

[Billard *et al.* 2008] Aude Billard, Sylvain Calinon, Rüdiger Dillmann and Stefan Schaal. *Robot Programming by Demonstration.* In Siciliano & Khatib [Siciliano & Khatib 2008], pages 1371–1394.

[Bitzer & van der Smagt 2006] S. Bitzer and P. van der Smagt. *Learning EMG control of a robotic hand: towards active prostheses.* In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, pages 2819 –2823, may 2006.

[Bullock *et al.* 2012] Ian M. Bullock, Raymond R. Ma and Aaron M. Dollar. *A Hand-Centric Classification of Human and Robot Dexterous Manipulation.* IEEE Transactions on Haptics, vol. 99, no. PrePrints, page 1, 2012.

[Buss 2004] Samuel R. Buss. *Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods.* Technical report, IEEE Journal of Robotics and Automation, 2004.

[Butterfass *et al.* 1998] J. Butterfass, G. Hirzinger, S. Knoch and H. Liu. *DLR's Multisensory Articulated Hand - Part I: Hard- and Software Architecture.* pages 2081–2086, 1998.

[Camarillo *et al.* 2004] David B. Camarillo, Thomas M. Krummel and J.Kenneth Salisbury Jr. *Robotic technology in surgery: Past, present, and future.* The American Journal of Surgery, vol. 188, no. 4, Supplement 1, pages 2 – 15, 2004.

[Chang *et al.* 2007] Lillian Y. Chang, Nancy Pollard, Tom Mitchell and Eric P. Xing. *Feature Selection for Grasp Recognition from Optical Markers.* In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pages 2944 – 2950, October 2007.

[Cherif & Gupta 1999] Moez Cherif and Kamal K. Gupta. *Planning quasi-static fingertip manipulations for reconfiguring objects*. IEEE Transactions on Robotics and Automation, vol. 15, no. 5, pages 837 –848, oct 1999.

[Cherif & Gupta 2001] Moez Cherif and Kamal K. Gupta. *Global Planning for Dexterous Reorientation of Rigid Objects: Finger Tracking with Rolling and Sliding*. The International Journal of Robotics Research, vol. 20, no. 1, pages 57–84, January 2001.

[Cutkosky 1989] M.R. Cutkosky. *On grasp choice, grasp models, and the design of hands for manufacturing tasks*. IEEE Transactions on Robotics and Automation, vol. 5, no. 3, pages 269–279, Jun 1989.

[Dejmal & Zacksenhouse 2006] I. Dejmal and M. Zacksenhouse. *Coordinative structure of manipulative hand-movements facilitates their recognition.* IEEE Transactions on Biomedical Engineering, vol. 53, no. 12 Pt 1, pages 2455–2463, 2006.

[DeLaurentis *et al.* 2000] K. J. DeLaurentis, C. Mavroidis and C. Pfeiffer. *Development of a Shape Memory Alloy Actuated Robotic Hand*. In Proceedings of International Conference on New Actuators, 2000.

[Dennis & Hannay 2005] M. R. Dennis and J. H. Hannay. *Geometry of Călugăreanu's theorem*. Royal Society of London Proceedings Series A, vol. 461, pages 3245–3254, October 2005.

[Desai & Volz 1989] R.S. Desai and R.A. Volz. *Identification and verification of termination conditions in fine motion in presence of sensor errors and geometric uncertainties*. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, pages 800 –807 vol.2, may 1989.

[Diftler *et al.* 2003] M.A. Diftler, C.J. Culbert, R.O. Ambrose, Jr. Platt R. and W.J. Bluethmann. *Evolution of the NASA/DARPA Robonaut control system*. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, volume 2, pages 2543 – 2548 vol.2, sept. 2003.

[Do *et al.* 2009] M. Do, J. Romero, H. KjellstrÃ¶m, P. Azad, T. Asfour, D. Kragic and R. Dillmann. *Grasp Recognition and Mapping on Humanoid Robots*. In Proceedings of the IEEE-RAS International Conference on Humanoid Robots, Humanoids, 2009.

[Ekvall & Kragic 2005] S. Ekvall and D. Kragic. *Grasp Recognition for Programming by Demonstration*. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, pages 748 – 753, april 2005.

[Elliott & K.J. 1984] J.M. Elliott and Connolly K.J. *A classification of manipulative hand movements*. Developmental medicine and child neurology, vol. 26, no. 3, pages 283–296, Jun 1984.

[Endo *et al.* 2011] T. Endo, H. Kawasaki, T. Mouri, Y. Ishigure, H. Shimomura, M. Matsumura and K. Koketsu. *Five-Fingered Haptic Interface Robot: HIRO III*. IEEE Transactions on Haptics, vol. 4, no. 1, pages 14 –27, j-feb. 2011.

[Faria *et al.* 2012] Diego R. Faria, Ricardo Martins, Jorge Lobo and Jorge Dias. *Extracting data from human manipulation of objects towards improving autonomous robotic grasping*. Robotics and Autononous Systems, vol. 60, no. 3, pages 396–410, March 2012.

[Farry *et al.* 1996] K.A. Farry, I.D. Walker and R.G. Baraniuk. *Myoelectric teleoperation of a complex robotic hand*. IEEE Transactions on Robotics and Automation, vol. 12, no. 5, pages 775 –788, oct 1996.

[Fischer *et al.* 1998] M. Fischer, P. van der Smagt and G. Hirzinger. *Learning techniques in a dataglove based telemanipulation system for the DLR hand*. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, volume 2, pages 1603 –1608 vol.2, may 1998.

[Garcia *et al.* 2007] E. Garcia, M.A. Jimenez, P.G. de Santos and M. Armada. *The evolution of robotics research*. IEEE Robotics and Automation Magazine, vol. 14, no. 1, pages 90 –103, march 2007.

[Griffin *et al.* 2000] W. B. Griffin, R. P. Findley, M. L. Turner and M. R. Cutkosky. *Calibration and mapping of a human hand for dexterous telemanipulation*. In ASME IMECE 2000 Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems, pages 1–8, 2000.

[Hagen & Dereniak 2008] Nathan Hagen and Eustace L. Dereniak. *Gaussian profile estimation in two dimensions*. Appl. Opt., vol. 47, no. 36, pages 6842–6851, Dec 2008.

[Han & Trinkle 1998] L. Han and J.C. Trinkle. *Dextrous manipulation by rolling and finger gaiting*. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, volume 1, pages 730 –735 vol.1, may 1998.

[Ho & Komura 2009a] Edmond S. L. Ho and Taku Komura. *Character Motion Synthesis by Topology Coordinates*. In P. Dutr'e and M. Stamminger, editors, Computer Graphics Forum (Proc. Eurographics 2009), volume 28, Munich, Germany, March 2009.

[Ho & Komura 2009b] Edmond S.L. Ho and Taku Komura. *Indexing and Retrieving Motions of Characters in Close Contact*. IEEE Transactions on Visualization and Computer Graphics, vol. 15, pages 481–492, 2009.

[Hu *et al.* 2004] Haiying Hu, Xiaohui Gao, Jiawei Li, Jie Wang and Hong Liu. *Calibrating human hand for teleoperating the HIT/DLR hand*. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, volume 5, pages 4571 – 4576 Vol.5, april-1 may 2004.

[Ikeuchi & Suehiro 1994] K. Ikeuchi and T. Suehiro. *Toward an assembly plan from observation. I. Task recognition with polyhedral objects*. IEEE Transactions on Robotics and Automation, vol. 10, no. 3, pages 368–385, Jun 1994.

[Jacobsen *et al.* 1984] S. C. Jacobsen, J. E. Wood, D. F. Knutti and K. B. Biggers. *The UTAH/M.I.T. Dextrous Hand: Work in Progress*. The International Journal of Robotics Research, vol. 3, no. 4, pages 21–50, December 1984.

[Johansson & Flanagan 2009] Roland S. Johansson and Randall J. Flanagan. *Coding and use of tactile signals from the fingertips in object manipulation tasks*. Nature Reviews Neuroscience, vol. 10, no. 5, pages 345–359, April 2009.

[Jones & Lederman 2006] L.A. Jones and S.J. Lederman. Human hand function. Oxford University Press, USA, 2006.

[Kamakura *et al.* 1980] N. Kamakura, M. Matsuo, H. Isshi, F. Mitsuboshi and Y. Miura. *Patterns of static prehension in normal hands*. The American journal of occupational therapy, vol. 34, no. 7, pages 437–445, Jul 1980.

[Kang & Ikeuchi 1993] Sing Bing Kang and K. Ikeuchi. *Toward automatic robot instruction from perception-recognizing a grasp from observation*. IEEE Transactions on Robotics and Automation, vol. 9, no. 4, pages 432–443, Aug 1993.

[Kang & Ikeuchi 1995] Sing Bing Kang and K. Ikeuchi. *Toward automatic robot instruction from perception-temporal segmentation of tasks from human hand motion*. IEEE Transactions on Robotics and Automation, vol. 11, no. 5, pages 670–681, Oct 1995.

[Kang & Ikeuchi 1997] Sing Bing Kang and K. Ikeuchi. *Toward automatic robot instruction from perception-mapping human grasps to manipulator grasps*. Robotics and Automation, IEEE Transactions on, vol. 13, no. 1, pages 81–95, Feb 1997.

[Kao *et al.* 2008] Imin Kao, Kevin Lynch and Joel W. Burdick. *Contact Modeling and Manipulation.* In Siciliano & Khatib [Siciliano & Khatib 2008], pages 647–669.

[Kemp *et al.* 2007] C.C. Kemp, A. Edsinger and E. Torres-Jara. *Challenges for robot manipulation in human environments [Grand Challenges of Robotics].* IEEE Robotics and Automation Magazine, vol. 14, no. 1, pages 20 –29, march 2007.

[Kjellstrom *et al.* 2008] H. Kjellstrom, J. Romero and D. Kragic. *Visual recognition of grasps for human-to-robot mapping.* In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pages 3192–3199, Sept. 2008.

[Klenin & Langowski 2000] K. Klenin and J. Langowski. *Computation of writhe in modeling of supercoiled DNA.* Biopolymers, vol. 54, pages 307–317, 2000.

[Kondo *et al.* 2006] Masahiro Kondo, Jun Ueda, Yoshio Matsumoto and Tsukasa Ogasawara. *Recognition of In-Hand Manipulation along with Rolling Contact using Orbital Motion of Contact Points on Object Surface.* In Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI, pages 167 –172, sept. 2006.

[Kondo *et al.* 2008] Masahiro Kondo, Jun Ueda and Tsukasa Ogasawara. *Recognition of in-hand manipulation using contact state transition for multifingered robot hand control.* Robotics and Autononous Systems, vol. 56, no. 1, pages 66–81, 2008.

[Kudoh *et al.* 2008] Shunsuke Kudoh, Naoto Ikeda, Koichi Ogawara and Katsushi Ikeuchi. *Learning everyday object manipulation from observation.* In IROS 2008 Workshop: Grasp and Task Learning by Imitation, pages 77 –82, sept. 2008.

[Lam *et al.* 1999] Pak Chio Lam, Yun-Hui Liu, Dazhai Li and M.Y.Y. Leung. *Generating dextrous manipulation for multifingered robot hands by combining motion planning and teaching.* vol. 3, pages 1419–1424 vol.3, 1999.

[Lii *et al.* 2010] N.Y. Lii, Zhaopeng Chen, B. Pleintinger, C.H. Borst, G. Hirzinger and A. Schiele. *Toward understanding the effects of visual- and force-feedback on robotic hand grasping performance for space teleoperation.* In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pages 3745 –3752, oct. 2010.

[Lovchik *et al.* 1999] C. S. Lovchik, Iovchikq Jsc. Nasa. Gov and M. A. Diftler. *The robonaut hand: A dexterous robot hand for space.* 1999.

[Martin *et al.* 2004] T.B. Martin, R.O. Ambrose, M.A. Diftler, Jr. Platt R. and M.J. Butzer. *Tactile gloves for autonomous grasping with the NASA/DARPA Robonaut.* In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, volume 2, pages 1713 – 1718 Vol.2, 26-may 1, 2004.

[Martins *et al.* 2010] Ricardo Martins, Diego R. Faria and Jorge Dias. *Symbolic level generalization of in-hand manipulation tasks from human demonstrations using tactile data information.* In IROS 2010 Workshop: Grasp Planning and Task Learning by Imitation, 2010.

[Maycock *et al.* 2011] Jonathan Maycock, Jan Steffen, Robert Haschke and Helge Ritter. *Robust tracking of human hand postures for robot teaching.* In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pages 2947–2952, 2011.

[Michelman 1998] P. Michelman. *Precision object manipulation with a multi-fingered robot hand.* IEEE Transactions on Robotics and Automation, vol. 14, no. 1, pages 105 –113, feb 1998.

[Miller & Allen 2004] A.T. Miller and P.K. Allen. *Graspit! A versatile simulator for robotic grasping.* IEEE Robotics and Automation Magazine, vol. 11, no. 4, pages 110 – 122, dec. 2004.

[Nakamura & Hanafusa 1986] Yoshihiko Nakamura and Hideo Hanafusa. *Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control.* Journal of Dynamic Systems, Measurement, and Control, vol. 108, no. 3, pages 163–171, 1986.

[Nakaoka *et al.* 2007] Shin'Ichiro Nakaoka, Atsushi Nakazawa, Fumio Kanehiro, Kenji Kaneko, Mitsuharu Morisawa, Hirohisa Hirukawa and Katsushi Ikeuchi. *Learning from Observation Paradigm: Leg Task Models for Enabling a Biped Humanoid Robot to Imitate Human Dances.* International Journal of Robotics Research, vol. 26, no. 8, pages 829–844, August 2007.

[Napier 1956] J. R. Napier. *The prehensile movements of the human hand.* The Journal of Bone and Joint Surgery, vol. 38-B, no. 4, pages 902–913, Nov 1956.

[Nocedal & Wright 2006] J. Nocedal and S.J. Wright. Numerical optimization. Springer series in operations research. Springer, 2006.

[Nolker & Ritter 2002] C. Nolker and H. Ritter. *Visual recognition of continuous hand postures.* IEEE Transactions on Neural Networks, vol. 13, no. 4, pages 983 – 994, jul 2002.

[Reichel 2004] Marco Reichel. *Transformation of Shadow Dextrous Hand and Shadow Finger Test Unit from Prototype to Product for Intelligent Manipulation and Grasping.* In Proceedings of International Conference Intelligent Manipulation and Grasping, July 2004.

[Saut *et al.* 2006] J.-P. Saut, A. Sahbani and V. Perdereau. *A Global Approach for Dexterous Manipulation Planning Using Paths in n-fingers Grasp Subspace.* In Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on, pages 1 –6, dec. 2006.

[Saut *et al.* 2007] J.-P. Saut, A. Sahbani, S. El-Khoury and V. Perdereau. *Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces.* In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pages 2907 –2912, 29 2007-nov. 2 2007.

[Scharfe *et al.* 2012] H. Scharfe, N. Hendrich and Jianwei Zhang. *Hybrid physics simulation of multi-fingered hands for dexterous in-hand manipulation.* In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, pages 3777 –3783, may 2012.

[Siciliano & Khatib 2008] Bruno Siciliano and Oussama Khatib, editors. Springer handbook of robotics. Springer, 2008.

[Steffen *et al.* 2007] Jan Steffen, Robert Haschke and Helge Ritter. *Experience-based and Tactile-driven Dynamic Grasp Control.* In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2007.

[Steffen *et al.* 2008] J. Steffen, R. Haschke and H. Ritter. *Towards dextrous manipulation using manipulation manifolds.* In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pages 2738 –2743, sept. 2008.

[Steffen *et al.* 2009] Jan Steffen, Stefan Klanke, Sethu Vijayakumar and Helge Ritter. *Realising Dextrous Manipulation with Structured Manifolds using Unsupervised Kernel Regression with Structural Hints.* In ICRA 2009 Workshop: Approaches to Sensorimotor Learning on Humanoid Robots, 2009.

[Steffen *et al.* 2010] J. Steffen, E. Oztop and H. Ritter. *Structured unsupervised kernel regression for closed-loop motion control.* In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pages 75 –80, oct. 2010.

[Steil *et al.* 2004] J.J. Steil, F. Rothling, R. Haschke and H. Ritter. *Situated robot learning for multi-modal instruction and imitation of grasping.* Robotics and Autonomous Systems, vol. 47, no. 2 - 3, pages 129 – 141, 2004.

[Sudsang & Phoka 2003] A. Sudsang and T. Phoka. *Regrasp planning for a 4-fingered hand manipulating a polygon.* In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, volume 2, pages 2671 – 2676 vol.2, sept. 2003.

[Suehiro & Ikeuchi 1992] T. Suehiro and K. Ikeuchi. *Towards An Assembly Plan From Observation: Part II: Correction Of Motion parameters Based On Fact Contact Constraints.* In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, volume 3, pages 2095 –2102, Jul 1992.

[Takamatsu *et al.* 2006] J. Takamatsu, T. Morita, K. Ogawara, H. Kimura and K. Ikeuchi. *Representation for knot-tying tasks.* IEEE Transactions on Robotics, vol. 22, no. 1, pages 65 – 78, feb. 2006.

[Takamatsu *et al.* 2007] Jun Takamatsu, Koichi Ogawara, Hiroshi Kimura and Katsushi Ikeuchi. *Recognizing Assembly Tasks Through Human Demonstration.* International Journal of Robotics Research, vol. 26, no. 7, pages 641–659, July 2007.

[Taylor 1985] Moore Taylor. *Robots for nuclear power plants.* In IAEA Bulletin, volume 27, 1985.

[Trinkle & Hunter 1991] J.C. Trinkle and J.J. Hunter. *A framework for planning dexterous manipulation.* In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, pages 1245 –1251 vol.2, apr 1991.

[Vinayavekhin *et al.* 2009] Phongtharin Vinayavekhin, Shunsuke Kudoh and Katsushi Ikeuchi. *Contact States Detection for Dexterous Manipulation in Low-Dimensional Joint Space.* In The 27th Annual Conference of The Robotics Society of Japan, September 2009.

[Vinayavekhin *et al.* 2011] Phongtharin Vinayavekhin, Shunsuke Kudoh and Katsushi Ikeuchi. *Towards an automatic robot regrasping movement based on human demonstration using tangle topology.* In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, pages 3332 –3339, may 2011.

[Wampler 1986] C.W. Wampler. *Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods.* IEEE

Transactions on Systems, Man, and Cybernetics, vol. 16, no. 1, pages 93 –101, jan. 1986.

[Witkin 1983] Andrew P. Witkin. *Scale-space filtering.* In Proceedings of the Eighth international joint conference on Artificial intelligence - Volume 2, IJCAI'83, pages 1019–1022, San Francisco, CA, USA, 1983. Morgan Kaufmann Publishers Inc.

[Yashima *et al.* 2003] M. Yashima, Y. Shiina and H. Yamaguchi. *Randomized manipulation planning for a multi-fingered hand by switching contact modes.* In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, volume 2, pages 2689 – 2694 vol.2, sept. 2003.

[Yousef *et al.* 2011] Hanna Yousef, Mehdi Boukallel and Kaspar Althoefer. *Tactile sensing for dexterous in-hand manipulation in robotics – A review.* Sensors and Actuators A: Physical, vol. 167, no. 2, pages 171 – 187, 2011. ¡ce:title¿Solid-State Sensors, Actuators and Microsystems Workshop¡/ce:title¿.

[Zacksenhouse & Moestl 1999] M. Zacksenhouse and T. Moestl. *Segmenting manipulative hand movements by dividing phase plane trajectories.* volume 3, pages 1910–1915 vol.3, 1999.

[Zhang *et al.* 1996] Hong Zhang, K. Tanie and H. Maekawa. *Dextrous manipulation planning by grasp transformation.* In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, volume 4, pages 3055 –3060 vol.4, apr 1996.

[Zollner *et al.* 2002] R. Zollner, O. Rogalla, R. Dillmann and M. Zollner. *Understanding users intention: programming fine manipulation tasks by demonstration.* In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, volume 2, pages 1114 – 1119 vol.2, 2002.